5 Công nghệ cơ sở dữ liệu

Mục đích

Cơ sở dữ liệu là một tập được tổ chức bao gồm các dữ liệu được thu thập lại nhằm mục đích chia sẻ dữ liệu, quản lý tập trung và có mức độ độc lập cao. Các cơ sở dữ liệu có nhiều loại nhưng hiện nay các loại chính bao gồm: cơ sở dữ liệu phân cấp, cơ sở dữ liệu mạng và cơ sở dữ liệu quan hệ. Trong đó cơ sở dữ liệu quan hệ đang là xu hướng chính hiện nay. Trong phần 1 chúng ta sẽ tìm hiểu về lý thuyết các cơ sở dữ liệu, thảo luận các cấu trúc của chúng và các phương pháp phát triển. Trong phần 2, chúng ta sẽ học cách sử dụng SQL – ngôn ngữ lập trình được dùng để thao tác cơ sở dữ liệu quan hệ. Cuối cùng, trong phần 3, chúng ta sẽ tìm hiểu DBMS – phần mềm phục vụ việc sử dụng hiệu quả các cơ sở dữ liệu.

5.1 Các mô hình dữ liệu
5.2 Các ngôn ngữ cơ sở dữ liệu
5.3 Thao tác cơ sở dữ liệu

[Các thuật ngữ và khái niệm cần nắm vững]

Lược đồ 3 lớp, cơ sở dữ liệu phân cấp, cơ sở dữ liệu mạng, cơ sở dữ liệu quan hệ, biểu đồ E-R, chuẩn hóa, phép chọn, phép chiếu, phép kết nối, DDL, DML, SQL, xác nhận 2 pha, nhân bản

5.1 Các mô hình dữ liệu

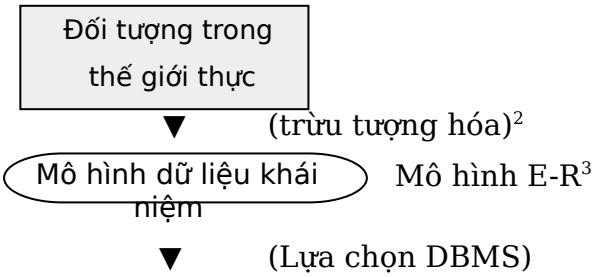
Mở đầu

Mô hình dữ liệu là sự mô tả khái quát các đối tượng dữ liệu có liên quan đến nhau theo một số luật nào đó. Để tạo ra một cơ sở dữ liệu đầu tiên phải đưa ra một mô hình dữ liệu sau đó chuẩn hóa nó bằng cách loại bỏ các thông tin không cần thiết và các đối tượng trùng lặp. Tiếp theo, cơ sở dữ liệu phải được thiết kế sao cho có thể tìm kiếm, cập nhật và xóa dữ liệu một cách hiệu quả. Các loại mô hình dữ liệu bao gồm: các mô hình dữ liệu khái niệm, các mô hình dữ liệu logic và các mô hình dữ liệu vật lý. Các luật để thực thi các mô hình trên gọi là: lược đồ khái niệm, lược đồ ngoài và lược đồ trong.

5.1.1 Lược đồ 3 lớp

Điểm chính	➢ **Khung (định nghĩa) của một cơ sở dữ liệu được gọi là lược đồ 3 lớp.** ➢ **Lược đồ 3 lớp bao gồm: lược đồ khái niệm, lược đồ ngoài và lược đồ trong.**

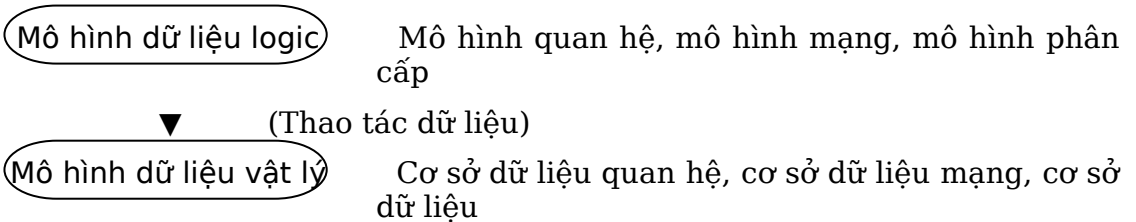
Việc trừu tượng hóa và tổ chức cấu trúc các thông tin thế giới thực (đối tượng để tạo thành một cơ sở dữ liệu) sau đó xử lý chúng được gọi là **mô hình hóa dữ liệu** *(data modeling).* Một mô hình dữ liệu có thể là mô hình khái niệm, logic hay vật lý. Mối liên hệ giữa chúng được chỉ ra trong hình vẽ dưới đây. [1]



[1] (Chú ý) Một mô hình dữ liệu là một sự biểu diễn khái niệm (mô hình) của dữ liệu; nó có thể cũng dẫn đến các luật biểu diễn.

[2] **Trừu tượng hóa**: Lấy ra những thành phần mô tả quan trọng nhất của đối tượng và loại bỏ những cái khác. Chúng ta có thể tạo ra một cơ sở dữ liệu để chia sẻ nếu như: khi tạo một mô hình dữ liệu đầu tiên phải tìm ra tất cả các thành phần phổ biến trong tất cả các thao tác từ các thao tác giữa các dữ liệu đến các thao tác cần được hệ thống hóa.

[3] (Gợi ý) Các mô hình khái niệm bao gồm cả mô hình E-R sẽ được thảo luận trong phần 5.1.3. Các mô hình dữ liệu logic trong đó có mô hình quan hệ, mô hình mạng và mô hình phân cấp. Hơn nữa, nếu chúng ta chọn một mô hình dữ liệu logic và tạo một cơ sở dữ liệu từ nó, ta có thể có được một mô hình dữ liệu vật lý.

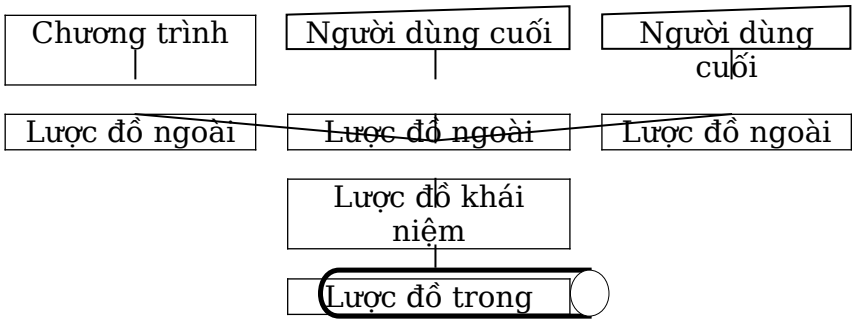

Mô hình dữ liệu logic
Mô hình quan hệ, mô hình mạng, mô hình phân cấp
▼
(Thao tác dữ liệu)
Mô hình dữ liệu vật lý
Cơ sở dữ liệu quan hệ, cơ sở dữ liệu mạng, cơ sở dữ liệu
phân cấp

Độc lập dữ liệu

Sự độc lập dữ liệu có thể hiểu là "Một chương trình sẽ không thay đổi khi dữ liệu thay đổi". Từ khi nhiều chương trình có thể chia sẻ một tập hợp dữ liệu chúng ta không cần phải tạo ra nhiều tập dữ liệu ứng với các chương trình riêng lẻ. Bởi thế dữ liệu cần phải được tổ chức một cách hệ thống.
Một loại phần mềm nhằm đạt được sự độc lập dữ liệu là hệ quản trị cơ sở dữ liệu (DBMS). DBMS đảm bảo sử độc lập dữ liệu bằng cách sử dụng *lược đồ 3 lớp*.

Lược đồ 3 lớp

Một lược đồ 3 lớp là sự mô tả khung của một cơ sở dữ liệu. Trong ANSIX3/SPARC,[4] các lược đồ được chia thành lược đồ khái niệm, lược đồ ngoài và lược đồ trong. Chúng được gọi là lược đồ 3 lớp. Quan hệ giữa chúng được biểu diễn trong hình vẽ sau đây:



Một cách tổng quát, người dùng sử dụng cơ sở dữ liệu qua lược đồ ngoài. [5]

Tên	Giải thích
Lược đồ ngoài	Định nghĩa cái nhìn cơ sở dữ liệu từ phía chương trình hoặc người dùng. Nó sử dụng một phần của lược đồ khái niệm. Trong các cơ sở dữ liệu quan hệ nó gọi là một "khung nhìn" còn trong các cơ sở dữ liệu mạng gọi là "lược đồ con". Nó tồn tại cho mỗi chương trình và người sử dụng.
Lược đồ khái niệm	Đây là dữ liệu được chứa trong cơ sở dữ liệu, định nghĩa dựa theo mô hình dữ liệu; một định nghĩa của dữ liệu thực nói chung. Nó được gọi là bảng trong cơ sở dữ liệu quan hệ và lược đồ trong cơ sở dữ liệu mạng.
Lược đồ ngoài	Đây là định nghĩa để thu được một lược đồ khái niệm dùng cho lưu trữ ngoài. Nó bao gồm các thông tin như là

[4] **ANSI/X3/SPARC**: ANSI (American National Standards Institute) là một tổ chức phi lợi nhuận đưa ra các chuẩn công nghiệp ở Mỹ. X3 là ủy ban trong ANSI thảo luận các chuẩn liên quan đến xử lý thông tin. SPARC (Standards Planning And Requirements Committee) là ủy ban liên quan đến các vấn đề quốc tế.
[5] (FAQ) Nhiều câu hỏi thi hỏi về các loại lược đồ và đặc điểm của nó. Cần hiểu được rõ ràng sự khác nhau giữa lược đồ khái niệm, lược đồ ngoài và lược đồ trong.

	môi trường, các phương pháp tổ chức và độ dài bộ đệm.

Liên quan đến cơ sở dữ liệu quan hệ và cơ sở dữ liệu mạng, xem phần 5.1.2 “ Các mô hình dữ liệu logic”.

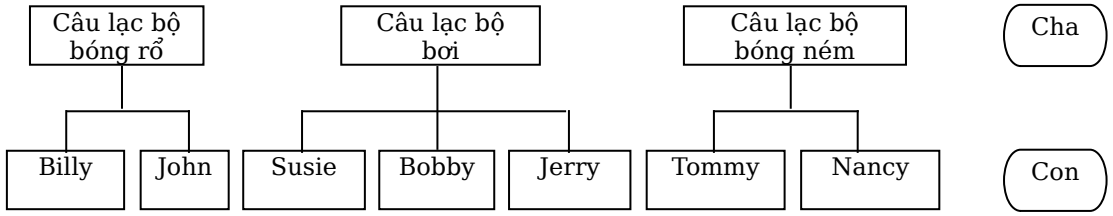
5.1.2 Các mô hình dữ liệu logic

Điểm chính	➢ **Các mô hình dữ liệu logic bao gồm mô hình quan hệ, mô hình mạng và mô hình phân cấp** ➢ **Một cơ sở dữ liệu là kết quả của việc thực thi một mô hình dữ liệu logic trong một môi trường lưu trữ**

Mô hình dữ liệu logic bao gồm **mô hình quan hệ, mô hình mạng và mô hình phân cấp**. Các mô hình dữ liệu này khi được thực thi sẽ trở thành **cơ sở dữ liệu quan hệ, cơ sở dữ liệu mạng** và **cơ sở dữ liệu phân cấp.** [6]

Cơ sở dữ liệu phân cấp (cơ sở dữ liệu cấu trúc dạng cây)

Một cơ sở dữ liệu phân cấp chia các bản ghi thành cha và con với mối liên hệ được thể hiện qua cấu trúc phân cấp. Nó được đặc trưng bởi mối liên hệ một-nhiều (1:n) giữa các bản ghi cha và các bản ghi con. Nói cách khác, một bản ghi cha có thể có nhiều bản ghi con nhưng một bản ghi con chỉ có thể có duy nhất một bản ghi cha. Tuy nhiên, các cơ sở dữ liệu phân cấp được đối xử như là một trường hợp đặc biệt của các cơ sở dữ liệu mạng, vì thế chúng không còn được sử dụng nhiều. Cấu trúc của một cơ sở dữ liệu phân cấp được chỉ ra như hình dưới đây:



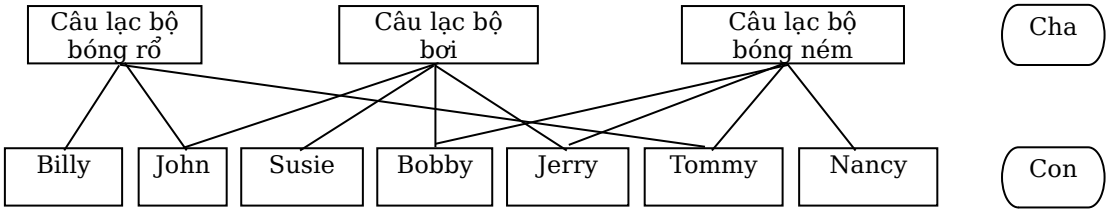
Cơ sở dữ liệu mạng

Một **cơ sở dữ liệu mạng** khác một cơ sở dữ liệu phân cấp ở chỗ các bản ghi cha và bản ghi con không chỉ có các mối liên hệ một-nhiều (1:n) mà chúng còn có các mối liên hệ nhiều-nhiều (m:n). Nói cách khác, một bản ghi cha có thể có nhiều bản ghi con và ngược lại, một bản ghi con cũng có thể có nhiều bản ghi cha.[7] Một cơ sở dữ liệu mạng đôi khi được gọi là một cơ sở dữ liệu CODASYL.[8] Cấu trúc của một cơ sở dữ liệu mạng được chỉ ra như hình dưới đây.

[6] (Chú ý) Các cơ sở dữ liệu phân cấp và cơ sở dữ liệu mạng đôi khi được gọi là các cơ sở dữ liệu cấu trúc (together)

[7] (Gợi ý) Một cơ sở dữ liệu cấu trúc là một cơ sở dữ liệu mạng trong đó mỗi nút con chỉ có duy nhất một nút cha

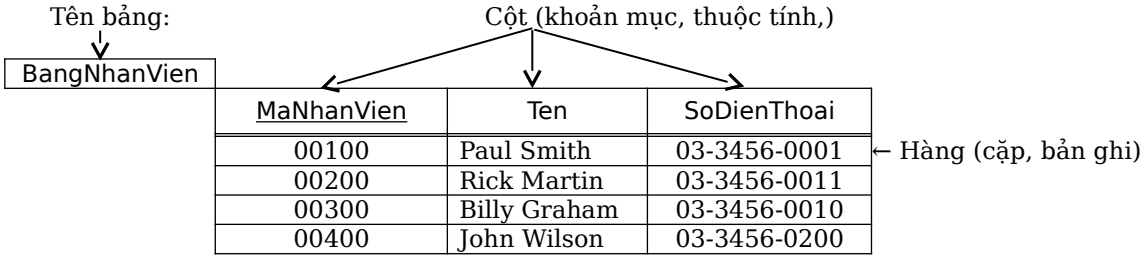
[8] **Cơ sở dữ liệu CODASYL**: Cơ sở dữ liệu mạng để cập đến bất kỳ một cơ sở dữ liệu nào dựa trên ngôn ngữ đặc trưng đưa ra bởi CODASYL; do vậy, một cơ sở dữ liệu mạng cũng được gọi là cơ sở dữ liệu CODASYL. CODASYL là viết tắt của Conference On DAta SYstems Languages. Tổ chức này bao gồm chính phủ Mỹ, các nhà sản xuất máy tính và người dùng. Đây là tổ chức đã phát triển và duy trì ngôn ngữ lập trình hướng kinh doanh COBOL. Nó đã phát triển COBOL năm 1960 và sau đó mới nghiên cứu về cơ sở dữ liệu.



Ví dụ ở đây “Susie” chỉ thuộc về “câu lạc bộ bơi” nhưng “Tommy” thuộc cả “câu lạc bộ bóng ném” và “câu lạc bộ bóng rổ.”

Cơ sở dữ liệu quan hệ

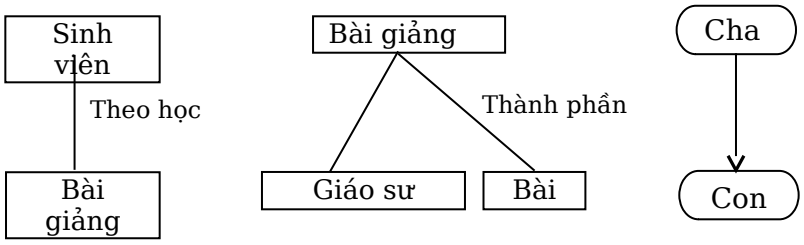
Cơ sở dữ liệu quan hệ là một cơ sở dữ liệu trong đó dữ liệu được biểu diễn trong một bảng 2 chiều. Mỗi hàng trong bảng ứng với một bản ghi và mỗi cột là một trường của bản ghi. Cột được gạch chân chỉ định khóa chính.[9]



Mỗi bảng luôn được đặt tên. Như ở ví dụ trên, tên của bảng là "BangNhanVien." Các cột là "MaNhanVien," "Ten," và "SoDienThoai" Mỗi hàng là một tập dữ liệu như "00100, Paul Smith, 03-3456-0001." Nói cách khác, chúng ta có thể nói rằng "BangNhanVien gồm có 4 hàng và 3 cột (*4 by 3*)."

Sơ đồ Bachman

Sơ đồ Bachman mô tả mối quan hệ cha-con giữa các bản ghi trong một cơ sở dữ liệu mạng. "Cha" gọi là chủ (*owner*) trong khi "con" gọi là thành viên (*member*). Bên dưới, các thuật ngữ như *theo học* (enrollment) và *thành phần* (component) mô tả các quan hệ cha-con gọi là các kiểu tập cha-con. Các nội dung thật sự (các giá trị) trong biểu đồ Bachman được gọi là **thể hiện**[10] (occurrence).



[9] **Khóa chính**: Là một cột hoặc một tập các cột xác định duy nhất một hàng trong bảng. Trong cùng một bảng, các giá trị khóa chính không thể trùng nhau. Ở đây trong "Employee table," "Employee number" là khóa chính. Nếu giá trị của một bảng không đơn nhất, có thể định nghĩa một khóa kết hợp bằng cách kết hợp nhiều cột.

[10] **Thể hiện**: Là một giá trị cụ thể trong sơ đồ Bachman. Ví dụ, nếu A, B và C là 3 "Sinh viên" thì A, B, C là 3 thể hiện.

5.1.3 Mô hình E-R và biểu đồ E-R

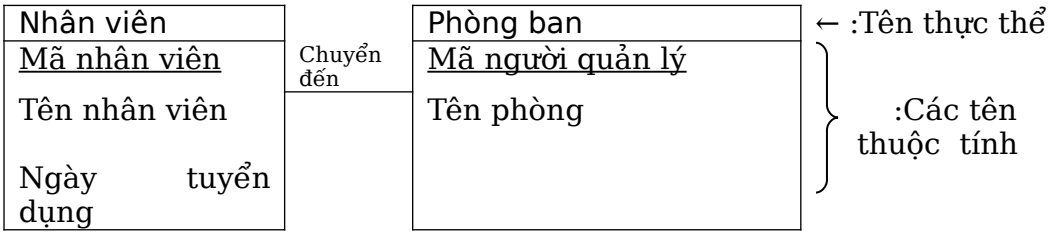
Điểm chính	➢ **Biểu đồ E-R (E-R diagram) là một mô hình dữ liệu không cần tính đến DBMS.** ➢ **Các thành phần của một biểu đồ E-R là "thực thể" (*entities*) và "quan hệ" (*relations*)**

Các mô hình phân cấp, mạng và quan hệ là tất cả các mô hình dữ liệu với giả dụ có sử dụng một DBMS. [11] Tuy nhiên, dữ liệu và thông tin sử dụng trong thế giới thực không nhất thiết phải giới hạn tương thích với DBMS. Một trong các phương pháp để biểu diễn các cấu trúc dữ liệu thế giới thực chính xác đến mức có thể là **mô hình thực thể liên kết E-R (E-R model)**. Mô hình E-R được biểu diễn bằng cách sử dụng một biểu đồ E-R.

Biểu đồ E-R là một kĩ thuật sử dụng trong thiết kế các tệp và cơ sở dữ liệu, dùng để biểu diễn các kết quả nhận được bởi việc thu thập các đối tượng để quản lý và các thành phần dữ liệu. Các đối tượng cần quản lý và phân tích là các thực thể, được liên kết với các thực thể khác bởi các quan hệ. Các yếu tố tạo thành các thực thể và quan hệ gọi là các thuộc tính.

Các quy tắc của biểu đồ E-R

Trong một biểu đồ E-R, các thực thể được biểu diễn trong các hộp hình chữ nhật trong khi các quan hệ được thể hiện bởi các đoạn thẳng và các mũi tên (←, —, →). Các thuộc tính cũng được biểu diễn trong các hộp. Trong ví dụ dưới đây, các Nhân viên và Phòng ban được liên kết bởi mối liên hệ có tên là "Chuyển đến". Thực thể "Nhân viên" có các thuộc tính là Mã nhân viên, tên nhân viên và Ngày tuyển dụng. Trong một số trường hợp các khóa chính được gạch chân.

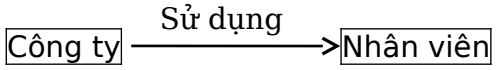


Các quan hệ thứ tự

Trong một biểu đồ E-R, quan hệ 1- nhiều "một công ty có nhiều nhân viên" được chỉ ra trong sơ đồ sau. Chú ý rằng, đôi khi không cần phải biểu diễn các thuộc tính. [12]

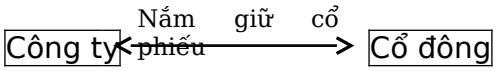
[11] **DBMS (Hệ quản trị cơ sở dữ liệu)**: Là phần mềm được thiết kế để duy trì và thi hành các cơ sở dữ liệu .

[12] (FAQ) Có nhiều câu hỏi thi hỏi giải thích các biểu đồ E-R. Chắc chắn rằng bạn có thể nhận ra các quan hệ 1- nhiều, 1-1 và nhiều-nhiều. Mối quan hệ giữa các nhân viên và các phòng ban trong một công ty với rất nhiều nhân viên, một số trong đó có thể thuộc về nhiều phòng ban, là "nhiều-



Ở đây, "Công ty" và "Nhân viên" được liên kết bởi mối quan hệ tên là "Sử dụng". Quan hệ giữa "Công ty" và "Nhân viên" là 1- nhiều (→) nên một công ty có nhiều nhân viên. Nếu một nhân viên được chọn, chỉ có duy nhất một công ty liên quan đến anh ta. Tuy nhiên, chọn một công ty thì không chỉ có duy nhất một nhân viên. Bởi thế với một quan hệ 1 - nhiều, chiều "1" là chiều xác định được.

Dưới đây là một biểu đồ E-R chỉ quan hệ nhiều – nhiều.



Ở đây có một mũi tên 2 chiều thể hiện mối liên hệ nhiều-nhiều giữa "Công ty" và "Cổ đông". Nó chỉ ra rằng một cổ đông có thể sở hữu cổ phiếu của nhiều công ty và một công ty cũng có thể có nhiều cổ đông. Nói cách khác, ta có thể hiểu hình vẽ trên như sau: "Có nhiều công ty, mỗi công ty có nhiều cổ đông".

5.1.4 Chuẩn hóa và các ràng buộc tham chiếu

Điểm chính	➢ **Chuẩn hóa có nghĩa là tối thiểu hóa sự dư thừa dữ liệu** ➢ **Có 3 dạng chuẩn hóa: dạng 1, dạng 2 và dạng 3**

➢ **Sự chuẩn hóa** (*Normalization*) - chuẩn hóa dữ liệu – là việc duy trì tính nhất quán và toàn vẹn dữ liệu bằng cách loại bỏ dư thừa dữ liệu . Có 3 dạng chuẩn hóa: dạng 1, dạng 2 và dạng 3. Sự chuẩn hóa là một khái niệm chỉ được sử dụng cho cơ sở dữ liệu quan hệ. Xu hướng chính hiện nay của cơ sở dữ liệu là cơ sở dữ liệu quan hệ nên chuẩn hóa cũng là một vấn đề rất quan trọng.

Dạng chưa chuẩn hóa

Dạng chưa chuẩn hóa là dạng mà các khoản mục chỉ được liệt kê một các đơn giản. Một cách tổng quát, các khoản mục trùng lặp cũng vẫn được bao hàm. Như trong hình vẽ dưới đây, một tổ hợp (Mã sản phẩm, Số lượng, Đơn giá) được lặp lại. Khoản mục được gạch chân là khóa chính. Ở đây Mã hóa đơn là khóa chính cho chúng ta biết rằng không thể có sự lặp lại Mã hóa

nhiều".

đơn. [13]

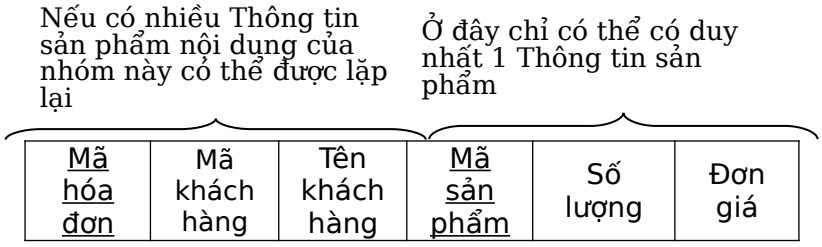
			Thông tin sản phẩm (1)			Thông tin sản phẩm (2)		
Mã hóa đơn	Mã khách hàng	Tên khách hàng	Mã sản phẩm	Số lượng	Đơn giá	Mã sản phẩm	Số lượng	Đơn giá

[13] (Gợi ý) Không thể có 2 bản ghi mà khóa chính có cùng giá trị. Trong ví dụ của dạng chưa chuẩn hóa này, khi Mã hóa đơn là khóa chính thì ở đây không có sự trùng lặp của Mã hóa đơn

Dạng chuẩn 1

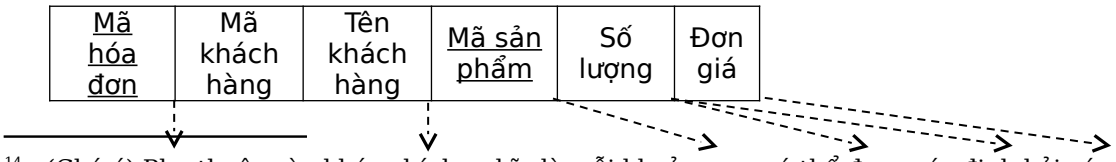
Dạng chuẩn 1 không cho phép sự lặp lại các khoản mục trong bảng. Để chuyển một dạng không chuẩn hóa về dạng chuẩn 1, chúng ta tách riêng tổ hợp (Mã sản phẩm, Số lượng, Đơn giá) trong dạng không chuẩn. Sau đó, thấy rằng có nhiều bản ghi trong đó (Mã hóa đơn, Mã khách hàng, Tên khách hàng) giống nhau, nhưng "Thông tin sản phẩm" lại khác nhau, bởi vậy "Mã hóa đơn" bản thân nó chưa thể làm thành khóa chính. Do đó, chúng ta có thể thêm "Mã sản phẩm" để cùng với "Mã hoá đơn"tạo thành khóa chính.

Trong ví dụ về dạng chuẩn ở trên, do có chứa sự lăp lại, vì vậy dạng chuẩn 1 chỉ có 2 bản ghi như chỉ ra sau đây:


Nếu có nhiều Thông tin sản phẩm nội dung của nhóm này có thể được lặp lại
Ở đây chỉ có thể có duy nhất 1 Thông tin sản phẩm
Mã hóa đơn
Mã khách hàng
Tên khách hàng
Mã sản phẩm
Số lượng
Đơn giá

Dạng chuẩn 2

Trong dạng chuẩn 1, hai khoản mục "Mã hóa đơn" và "Mã sản phẩm" cùng nhau làm thành khóa chính. [14] Trong các cơ sở dữ liệu, tất cả các thuộc tính không phải là khóa đều phải phụ thuộc hàm vào thực thể khóa chính. Tuy nhiên, "Đơn giá" không có quan hệ với "Mã hóa đơn". Khoản mục "Đơn giá" chỉ phụ thuộc duy nhất vào "Mã sản phẩm". Do đó chúng ta có thể tách riêng "Đơn giá". Để làm được điều này do "Mã sản phẩm" và "Đơn giá" phải được tương ứng với nhau chúng ta sử dụng "Mã sản phẩm" như là khóa chính. Hơn nữa, "Mã khách hàng" cũng có thể được xác định duy nhất nếu chọn "Mã hóa đơn", do đó "Tên khách hàng" là không cần thiết. Lúc này để xác định "Số lượng", chúng ta phải có "Mã hóa đơn" và "Mã sản phẩm". Cho các giải thích tiếp sau, chúng ta đặt tên các bảng mới là "Bảng hóa đơn", "Bảng chi tiết" và "Bảng sản phẩm". Kết quả của việc chia 3 bảng này gọi là **dạng chuẩn 2**. [15]



[14] (Chú ý) Phụ thuộc vào khóa chính nghĩa là mỗi khoản mục có thể được xác định bởi các giá trị khóa chính.

[15] **Phụ thuộc hàm đầy đủ/Phụ thuộc hàm bộ phận**: Trong dạng chuẩn 1, số lượng được xác định cho khóa chính "Mã hóa đơn + Mã sản phẩm". Như thấy ở ví dụ trên, sự phụ thuộc vào kết hợp của các khóa chính gọi là phụ thuộc hàm đầy đủ. Bên cạnh đó, Đơn giá chỉ phụ thuộc vào duy nhất một trong các khóa chính (ở đây là mã sản phẩm); Khi một khoản mục phụ thuộc vào duy nhất 1 khóa chính thì gọi là phụ thuộc hàm bộ phận. Một dạng chuẩn 2 có thể được định nghĩa "Dạng chuẩn 1 mà trong đó tất cả các thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào khóa chính".

Mã hóa đơn	Mã khách hàng	Tên khách hàng

(Bảng hóa đơn)

Mã hóa đơn	Mã sản phẩm	Số lượng

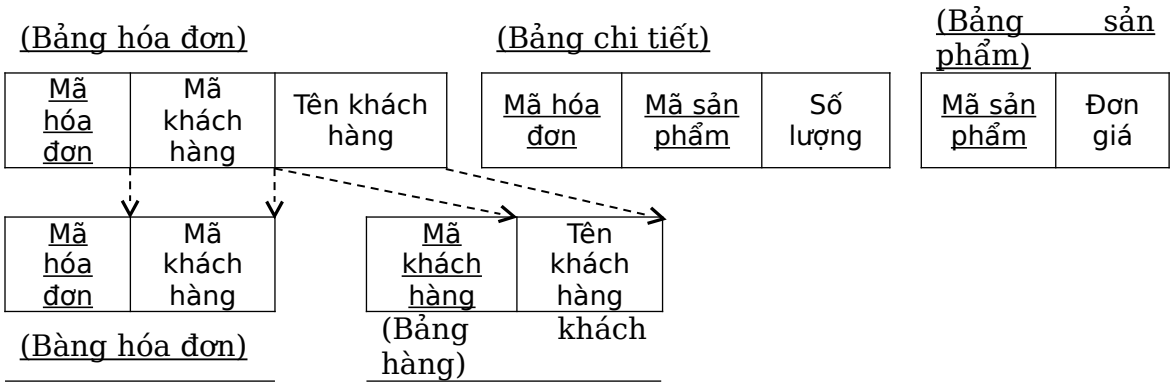
(Bảng chi tiết)

Mã sản phẩm	Đơn giá

(Bảng sản phẩm)

Dạng chuẩn 3

Chú ý rằng trong dạng chuẩn 2, trong "bảng hóa đơn" nói riêng, "Mã hóa đơn" xác định duy nhất "Mã khách hàng". Hơn nữa, khoản mục "Mã khách hàng" cũng xác định duy nhất "Tên khách hàng". Do đó, như chỉ ra dưới đây, chúng ta tách riêng "Tên khách hàng" và chuẩn bị một bảng mới "Bảng khách hàng", trong đó "Mã khác hàng" là khóa chính. Ở đây chúng ta không cần thay đổi gì với 2 bảng "Bảng chi tiết" và "Bảng sản phẩm".



Bởi vậy khi các bộ của dạng chưa chuẩn hóa được sửa đổi thành dạng chuẩn 3, các dữ liệu được chia thành 4 bộ: "Bảng chi tiết", "Bảng sản phẩm", "Bảng hóa đơn" và "Bảng khách hàng". Đặc tính của dạng chuẩn 3 là không có khoản mục nào được lặp lại ngoại trừ các khoản mục làm khoá chính. [16]

Các ràng buộc tham chiếu

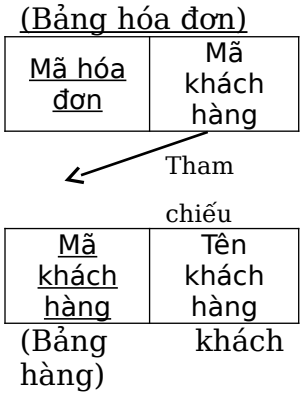
Nếu như không có sự mâu thuẫn giữa dữ liệu lưu trữ trong một cơ sở dữ liệu, chúng ta nói rằng cơ sở dữ liệu đó **nhất quán** *(consistent)*. Các điều kiện khác nhau nhằm kiểm tra sự toàn vẹn của dữ liệu gọi là **ràng buộc toàn vẹn** *(integrity constraints).* Các ràng buộc nhất quán bao gồm **ràng buộc tham chiếu *(reference constraints)*, ràng buộc tồn tại *(existence constraints)*, ràng buộc cập nhật *(update constraints)* và ràng buộc định dạng *(format constraints)*.** [17] [18]

[16] **Phụ thuộc hàm bắc cầu**: Tên khách hàng ở dạng chuẩn 2 có thể được xác định vì khóa chính Mã hóa đơn xác định Mã khách hàng do đó xác định Tên khách hàng. Nói cách khác Mã hóa đơn xác định gián tiếp Tên khách hàng. Dạng xác định gián tiếp này gọi là Phụ thuộc hàm bắc cầu.
Nói một cách chặt chẽ hơn, một dạng chuẩn 3 có thể được định nghĩa như sau: "Một dạng chuẩn 2 trong đó không có thuộc tính không khóa nào phụ thuộc bắc cầu vào khóa chính".

[17] (FAQ) Một số câu hỏi thi đưa ra các bản ghi ở một dạng không chuẩn cũng như là một vài giả định sau đó hỏi bạn tìm dạng chuẩn 3 trong một nhóm câu trả lời. Nếu bạn làm theo các bước như mô tả trong cuốn sách này để có được dạng chuẩn 3 bạn sẽ tìm được Đáp án đúng nhưng rất mất thời gian. Vì thế, ta nên tìm dạng chuẩn 3 bằng trực quan. Bạn có thể thử nhiều câu hỏi để thực hành nhưng bạn có thể xác định được dạng chuẩn 3 bằng các đặc điểm "không có sự trùng lặp các khoản mục trừ xác khoản mục khóa chính".

[18] **Ràng buộc tồn tại**: Nó có nghĩa là sự tồn tại của một dữ liệu đơn lẻ yêu cầu sự tồn tại của một

Ràng buộc tham chiếu là một ràng buộc liên quan đến sự nhất quán giữa các khoản mục trùng lặp. Nếu một bảng bao gồm dữ liệu được tra cứu từ một bảng khác thì bảng khác ấy phải có dữ liệu đăng ký tham chiếu từ trước đó. Ví dụ, trong dạng chuẩn 3 đã giải thích ở trên, để đăng ký "Bảng hóa đơn", trong "Bảng khách hàng" cần đăng ký thông tin về việc *mã khách hàng* tương thích với *mã khách hàng* trong "Bảng hóa đơn" . Ở đây, Mã khách hàng trong "Bảng hóa đơn" là một khóa ngoài của "Bảng Khách hàng".



(Mã khách hàng trong "Bảng hóa đơn" là một khóa ngoài của "Bảng Khách hàng")

5.1.5 Thao tác dữ liệu trong cơ sở dữ liệu quan hệ

Điểm chính	➢ **Thao tác dữ liệu bao gồm các phép toán quan hệ và các phép toán tập hợp** ➢ **Các phép toán quan hệ gồm các phép chọn, chiếu và kết nối**

Trong rất nhiều loại các phép toán trên cơ sở dữ liệu quan hệ thì các phép toán quan hệ và các phép toán tập hợp là quan trọng nhất. Trong một cơ sở dữ liệu quan hệ một bảng, một hàng và một cột đều được đối xử như những tập hợp có thể trích rút ra các giá trị. Các quá trình trích rút ra các giá trị này bao gồm sự thao tác như **chọn, chiếu và kết nối**. Chúng được gọi là các phép toán quan hệ. Trong khi đó, có nhiều những phép toán khác nhờ đó 2 bảng trong cơ sở dữ liệu quan hệ được sử dụng để tạo thành 1 bảng mới gọi là các phép toán tập hợp. Các phép toán tập hợp bao gồm phép **hợp, giao và trừ**.

---

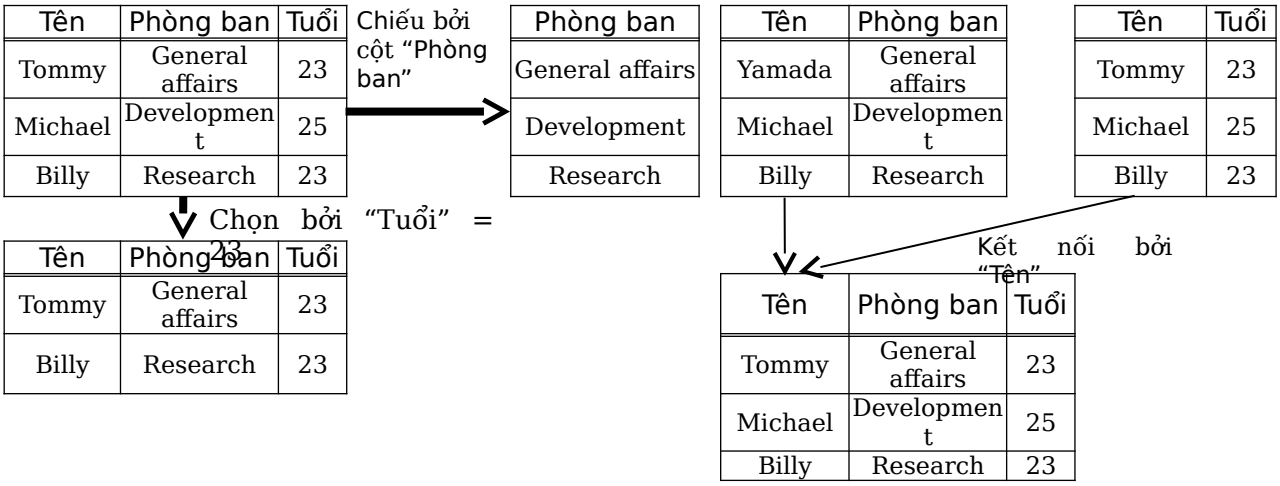
vài dữ liệu khác. Ví dụ, một bản ghi con không thể thêm vào trừ khi một bản ghi cha đã tồn tại.
Ràng buộc cập nhật: Nó có nghĩa là một khoản mục mới phải thỏa mãn các điều kiện đưa ra để có thể đăng ký hợp lệ. Ví dụ, giá trị "6" không thể đăng ký nếu giá trị phải nằm trong khoảng từ 1 đến 5.
Ràng buộc định dạng: Nó có nghĩa là một khoản mục phải có một định dạng thỏa mãn các điều kiện đưa ra. Ví dụ, văn bản không thể đăng ký trong một khoản mục yêu cầu giá trị số.

Các phép toán quan hệ

Ý nghĩa của các phép toán quan hệ được liệt kê trong bảng dưới đây. Rất nhiều dữ liệu được trích rút bằng cách kết hợp các phép toán cơ bản.

Phép toán	Chức năng
Chọn	Chọn ra các hàng thỏa mãn điều kiện nào đó
Chiếu	Rút ra một số cột đặc biệt (các thuộc tính)
Kết nối	Liên kết các bảng có các cột tương đương

Phép chiếu rút ra một cột đặc biệt nào đó. Ví dụ trong hình vẽ dưới đây, chỉ "Phòng ban" được rút ra. Phép chọn chọn ra một số hàng nào đó, ví dụ mỗi hàng mà "Tuổi" là 23 đều được chọn. Phép kết nối liên kết các cột tương đương, ví dụ 2 bảng được kết nối bởi "Tên" [19] [20]



Phép chọn cũng có thể chọn ra nhiều cột khác nhau.

Các phép toán tập hợp

Các **phép toán tập hợp** được dựa trên lý thuyết tập hợp. Bao gồm: [21]

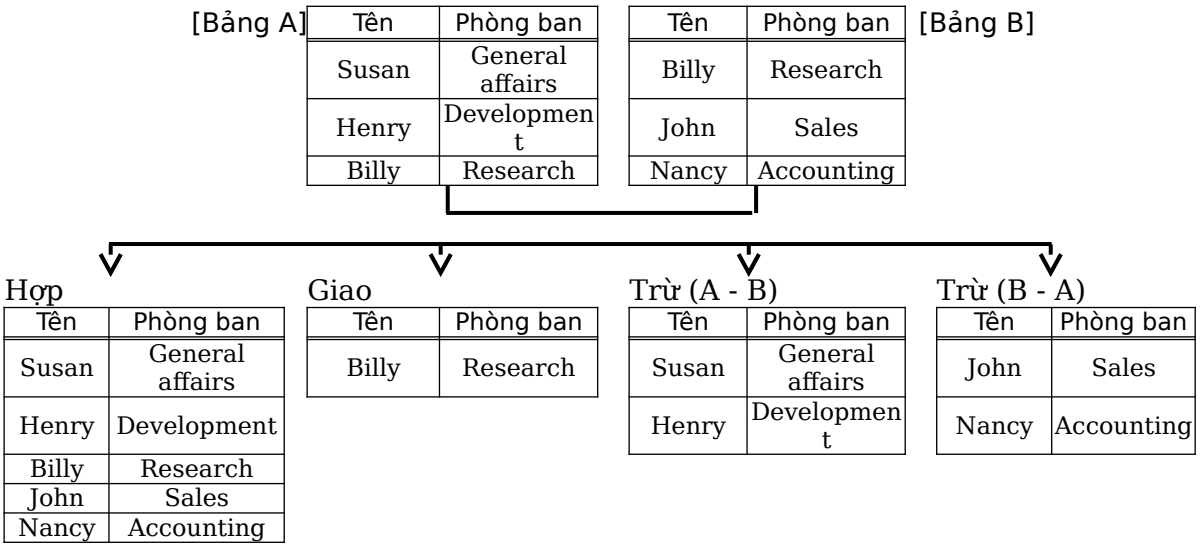
[19] (Gợi ý) Các kết quả của các phép toán quan hệ và tập hợp được biểu diễn như các bảng mới nhưng không thật sự lưu trữ trong cơ sở dữ liệu. Nó đơn giản lưu trữ trong vùng làm việc như những kết quả tức thời.

[20] (FAQ) Rất nhiều các câu hỏi thi bao hàm ý nghĩa của các phép toán quan hệ. Chắc chắn rằng bạn biết phép chọn lấy ra các "hàng" và phép chiếu lấy ra các "cột". Chắc chắn rằng bạn biết phép kết nối là một phép toán kết hợp nhiều bảng.

[21] **Sắp xếp/ 4 phép toán cơ bản**: Một cơ sở dữ liệu quan hệ không chỉ có các phép toán quan hệ và tập hợp mà có cả các phép toán sắp xếp và 4 phép toán cơ bản. Sắp xếp là hàm sắp xếp dữ liệu tăng hoặc giảm dần theo một cột nào đó. 4 phép toán cơ bản áp dụng cho các thuộc tính số và lấy kết quả bằng việc thao tác các biểu thức toán học (4 phép cơ bản). Ví dụ, có thể lấy kết quả của việc nhân giá trị của một cột nào đó với 10.

Phép toán	Chức năng
Hợp	Chọn ra các hàng nằm trong ít nhất 1 trong 2 bảng
Giao	Chọn ra các hàng chứa các giá trị như nhau ở tất cả các bảng
Trừ	Chọn ra các hàng nằm trong bảng này nhưng không nằm trong bảng khác

Ví dụ, phép hợp chọn các hàng xuất hiện trong Bảng A hoặc Bảng B; chú ý rằng "Billy" xuất hiện trong cả 2 bảng nên nó chỉ được chọn duy nhất 1 lần. Phép giao chọn lấy các hàng mà xuất hiện trong các Bảng A và Bảng B. Trong trường hợp này, chỉ "Billy" được chọn.
Thứ tự của các phép toán không thành vấn đề trong các phép hợp và giao, nhưng với phép trừ thứ tự lại là vấn đề. Các thứ tự khác nhau sẽ đưa đến các kết quả khác nhau. "A – B" đưa ra các hàng nằm trong A nhưng không nằm trong B. Ở đây, "Billy" bị bỏ qua nhưng "Susan" và "Henry" được chọn. Ngược lại, "B – A" đưa ra các hàng nằm trong B nhưng không nằm trong A, vì thế "Billy" lại một lần nữa bị bỏ qua, kết quả trả về là "John" và "Nancy" được chọn.



Câu hỏi nhanh

Q1 Liệt kê lược đồ 3 lớp và giải thích vai trò của từng lược đồ.

Q2 Liệt kê các mô hình dữ liệu logic và giải thích ngắn gọn đặc điểm của từng loại.

Q3 Mô tả các đặc điểm của dạng chuẩn 1, 2 và 3.

Q4 Liệt kê các loại phép toán quan hệ và giải thích cách thực hiện của từng loại.

A1 Lược đồ ngoài: *Định nghĩa cái nhìn cơ sở dữ liệu từ chương trình hoặc người dùng. Nó sử dụng một phần của lược đồ*

khái niệm. Trong các cơ sở dữ liệu quan hệ, đây gọi là một "khung nhìn"; trong các cơ sở dữ liệu mạng, gọi là "lược đồ con". Nó tồn tại cho mỗi chương trình và người sử dụng.

Lược đồ khái niệm: *Đây là dữ liệu được chứa trong cơ sở dữ liệu, định nghĩa dựa theo mô hình dữ liệu; một định nghĩa của dữ liệu thực nói chung. Nó được gọi là bảng trong cơ sở dữ liệu quan hệ và lược đồ trong cơ sở dữ liệu mạng*

Lược đồ trong: *Đây là định nghĩa việc thu được một lược đồ khái niệm cho sự lưu trữ ngoài. Nó bao gồm các thông tin như là tối thiểu, các phương pháp tổ chức và độ dài bộ đệm.*

A2 Mô hình phân cấp: Mối liên hệ giữa các cha và con là 1:n.
Mô hình mạng: Mối liên hệ giữa cha và con là m:n.
Mô hình quan hệ: Có dạng bảng

A3 Dạng chuẩn 1: Các trùng lặp được loại bỏ.
Dạng chuẩn 2: Đã ở dạng chuẩn 1 và các phụ thuộc hàm bộ phận được loại bỏ

Dạng chuẩn 3: Đã ở dạng chuẩn 2 và các phụ thuộc hàm bắc cầu được loại bỏ

A4 Phép chiếu: Chọn ra các cột nào đó (các thuộc tính)
Phép chọn: Chọn ra các hàng thỏa mãn điều kiện nào đó
Kết nối: Liên kết nhiều bảng có các cột tương đương

5.2 Các ngôn ngữ cơ sở dữ liệu

Mở đầu

Một ngôn ngữ cơ sở dữ liệu là một ngôn ngữ dùng để định nghĩa, xóa các cơ sở dữ liệu, các bảng, đồng thời tìm kiếm và cập nhật dữ liệu.

5.2.1 DDL và DML

Điểm chính	➢ **Các ngôn ngữ cơ sở dữ liệu bao gồm DDL và DML.** ➢ **SQL là một ngôn ngữ cơ sử dữ liệu dùng cho cơ sử dữ liệu quan hệ.**

Một ngôn ngữ dùng để định nghĩa và tổ chức các cơ sở dữ liệu gọi là một **ngôn ngữ định nghĩa dữ liệu** *(data definition language - DDL)* trong khi một ngôn ngữ dùng để tìm kiếm, cập nhật, thêm, xóa dữ liệu gọi là một ngôn ngữ thao tác dữ liệu *(data manipulation language - DML)*. Tổng quát, người quản trị cơ sở dữ liệu sử dụng DLL để chuẩn bị cơ sở dữ liệu trong khi người lập trình viên hệ thống lại sử dụng DML để phát triển các hệ thống sử dụng cơ sở dữ liệu. Các ngôn ngữ cơ sở dữ liệu đặc trưng bao gồm: SQL dùng cho cơ sở dữ liệu quan hệ và NDL[22] dùng cho các cơ sở dữ liệu mạng.

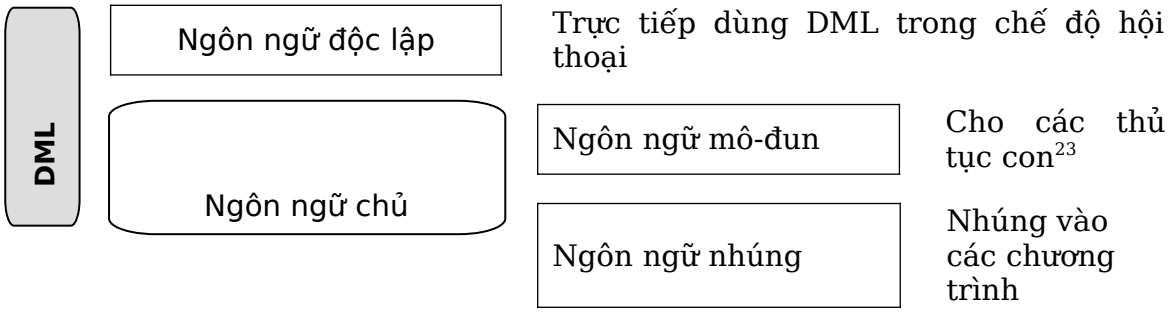
DDL (Ngôn ngữ định nghĩa dữ liệu)

DDL là một ngôn ngữ hệ thống dùng để định nghĩa các lược đồ dựa trên mô hình dữ liệu. Chúng được quy định là SQL-DDL cho các cơ sở dữ liệu quan hệ.

DML (Ngôn ngữ thao tác dữ liệu)

DML là một ngôn ngữ hệ thống dùng để các thao tác các cơ sở dữ liệu dành cho các người dùng. Chúng được quy định là SQL-DML cho các cơ sơ dữ liệu quan hệ. Dựa theo cách sử dụng, các DML được phân lớp như dưới đây:

[22] **NDL (Ngôn ngữ cơ sở dữ liệu mạng)**: Đây là ngôn ngữ cơ sở dữ liệu cho các cơ sở dữ liệu mạng, sử dụng để định nghĩa lược đồ và thao tác các cơ sở dữ liệu. NDL bao gồm các chức năng sau: ngôn ngữ định nghĩa lược đồ để định nghĩa cấu trúc của cơ sở dữ liệu; ngôn ngữ định nghĩa lược đồ con để định nghĩa các khung nhìn; ngôn ngữ thao tác dữ liệu để thao tác dữ liệu trong cơ sở dữ liệu và ngôn ngữ mô-đun để thực thi các thủ tục từ nhiều ngôn ngữ thao tác dữ liệu.


DML
Ngôn ngữ độc lập
Trực tiếp dùng DML trong chế độ hội thoại
Ngôn ngữ chủ
Ngôn ngữ mô-đun
Cho các thủ tục con[23]
Ngôn ngữ nhúng
Nhúng vào các chương trình

Ngôn ngữ độc lập

Ở đây ta đề cập đến một hệ thống cung cấp một ngôn ngữ lập trình khác với các ngôn ngữ thông thường sao cho các hàm cung cấp bởi hệ thống quản lý dữ liệu có thể được sử dụng trong phạm vi của các hàm trong ngôn ngữ mà nó cung cấp. Nó sử dụng SQL và NDL dưới dạng đối thoại như những câu lệnh.

Ngôn ngữ chủ

Ở đây đề cập đến một hệ thống thao tác dữ liệu ở đó DML được nhúng vào trong chương trình viết bằng ngôn ngữ cấp cao hơn như COBOL, Fortran, và C. Các ngôn ngữ cấp cao hơn gọi là các ngôn ngữ chủ.
Các phương pháp nhúng DML vào trong một chương trình bao gồm hệ thống ngôn ngữ mô-đun và hệ thống nhúng. Trong hệ thống ngôn ngữ mô-đun, chúng ta có thể phát triển một thủ tục con mà tạo thành các phần thao tác cơ sở dữ liệu của chương trình, và chương trình gọi thủ tục con bằng một lệnh "gọi". Không giống như thế, ngôn ngữ con nhúng (*embedded sublanguage*) là nơi DML được viết trực tiếp trong chương trình.

Hàm con trỏ

Hàm con trỏ (*cursor function*) được sử dụng khi xử lý các hàng (các bản ghi) của một cơ sở dữ liệu sử dụng một ngôn ngữ thủ tục. Nó xem kết quả truy vấn (bảng thu được) bởi DML như một tệp sao cho có thể xử lý bằng một ngôn ngữ lập trình.

Với hàm con trỏ, các tệp sử dụng bởi chương trình hiện tại có thể chuyển sang cơ sở dữ liệu dễ dàng. Trong SQL, có các câu lệnh thao tác sau. [24]

Câu lệnh thao tác	Ý nghĩa

[23] **Ngôn ngữ mô-đun**: Là ngôn ngữ dùng để viết trong một ngôn ngữ thao tác dữ liệu; nó xử lý các cơ sở dữ liệu khi được gọi từ một ngôn ngữ cấp cao hơn chẳng hạn như COBOL.
[24] (Chú ý) Trong DECLARE CURSOR, tên con trỏ được định nghĩa. Tiếp theo DECLARE CURSOR, một câu lệnh SELECT được viết (là một câu truy vấn viết bằng DML). Thủ tục này đưa ra một bảng kết quả. Sau đó, bằng lệnh FETCH, bảng bắt đầu được đọc từ hàng đầu tiên.

DECLARE CURSOR	Khai báo con trỏ
OPEN CURSOR	Bắt đầu hoạt động của con trỏ
FETCH	Đọc một hàng
CLOSE CURSOR	Kết thúc hoạt động của con trỏ

5.2.2 SQL

Điểm chính	➢ Câu lệnh dùng để trích rút dữ liệu là “SELECT.” ➢ Trong các câu truy vấn con làm rõ “SELECT” bằng cách sử dụng một mệnh đề WHERE

Ở đây chúng ta xem xét một cách chi tiết SQL – ngôn ngữ cơ sở dữ liệu cho các cơ sở dữ liệu quan hệ. Trong SQL, các câu lệnh SELECT dùng để trích rút dữ liệu từ một cơ sở dữ liệu.

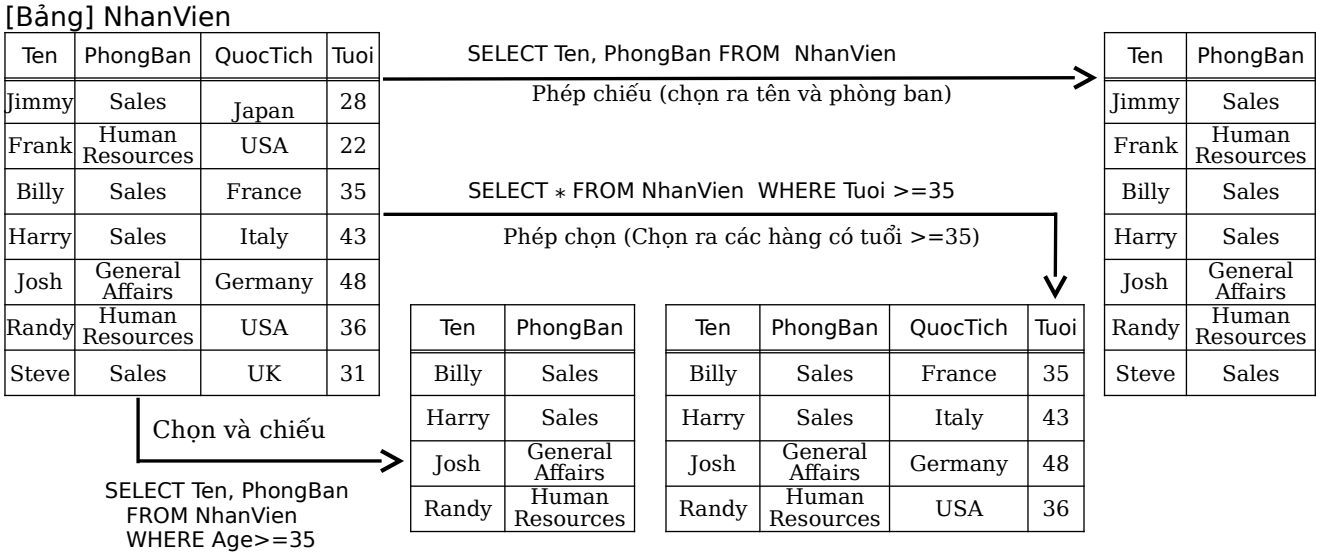
Cấu trúc của câu lệnh SELECT

Cấu trúc một câu lệnh SELECT như sau: [25]

```
SELECT    khoản mục 1, khoản mục 2, …
FROM      bảng 1, bảng 2, …
WHERE     điều kiện
```

Phép chiếu và phép chọn

Dưới đây là một ví dụ chỉ ra kết quả của việc trích rút dữ liệu bằng các phép chiếu và phép chọn sử dụng một câu lệnh SELECT trong bảng “Nhân viên”. Nếu theo sau SELECT là “*” tất cả các hàng trong bảng đều sẽ được chọn. Chúng ta cũng có thể kết hợp phép chọn và phép chiếu.


[Bảng] NhanVien
SELECT Ten, PhongBan FROM NhanVien
Phép chiếu (chọn ra tên và phòng ban)
SELECT * FROM NhanVien WHERE Tuoi >=35
Phép chọn (Chọn ra các hàng có tuổi >=35)
Chọn và chiếu
SELECT Ten, PhongBan
FROM NhanVien
WHERE Age>=35

Ten	PhongBan	QuocTich	Tuoi
Jimmy	Sales	Japan	28
Frank	Human Resources	USA	22
Billy	Sales	France	35
Harry	Sales	Italy	43
Josh	General Affairs	Germany	48
Randy	Human Resources	USA	36
Steve	Sales	UK	31

Ten	PhongBan
Jimmy	Sales
Frank	Human Resources
Billy	Sales
Harry	Sales
Josh	General Affairs
Randy	Human Resources
Steve	Sales

Ten	PhongBan
Billy	Sales
Harry	Sales
Josh	General Affairs
Randy	Human Resources

Ten	PhongBan	QuocTich	Tuoi
Billy	Sales	France	35
Harry	Sales	Italy	43
Josh	General Affairs	Germany	48
Randy	Human Resources	USA	36

[25] (Gợi ý) Trong một câu lệnh SELECT, mệnh đề WHERE có thể được bỏ qua. Nếu bỏ qua, điều kiện để lấy dữ liệu là không có nên tất cả các khoản mục được yêu cầu trong SELECT đều được chọn.

Kết nối

Với câu lệnh SELECT chúng ta có thể kết nối các bảng thông qua các cột đặc biệt nào đó. Dưới đây là một ví dụ về việc kết nối bảng ”Nhân viên” và bảng “Phòng ban”. Không thể có các cột trùng tên trong một bảng nhưng nếu có các cột tên giống nhau trong các bảng khác nhau thì chúng có thể phân biệt bằng cách dùng dạng **“TênBảng.TênCột”**. “NhanVien.PhongBan=PhongBan.PhongBan” là khóa kết nối 2 bảng này.

[Bảng] NhanVien

Ten	Phong	QuocTich	Tuoi
Jimmy	Sales	Japan	28
Frank	Human Resources	USA	22
Billy	Sales	France	35
Harry	Sales	Italy	43
Josh	General Affairs	Germany	48
Randy	Human Resources	USA	36
Steve	Sales	UK	31

[Bang] PhongBan

Phong	TruongPhong	ViTri
Human Resources	Randy	Headquarters
General Affairs	Josh	Headquarters
Sales	Harry	Branch office

```
SELECT Ten,
  NhanVien.PhongBan,
  QuocTich, Tuoi, TruongPhong,
  ViTri
FROM NhanVien,
  PhongBan
WHERE NhanVien.PhongBan
  =PhongBan.PhongBan[26]
```

Ten	Phong	QuocTich	Tuoi	TruongPhong	ViTri
Jimmy	Sales	Japan	28	Harry	Branch office
Frank	Human Resources	USA	22	Randy	Headquarters
Billy	Sales	France	35	Harry	Branch office
Harry	Sales	Italy	43	Harry	Branch office
Josh	General Affairs	Germany	48	Harry	Headquarters
Randy	Human Resources	USA	36	Randy	Headquarters
Steve	Sales	UK	31	Harry	Branch office

IN và BETWEEN

Trong mệnh đề **WHERE** chúng ta có thể kết hợp nhiểu điều kiện với **AND** và **OR**. IN chỉ ra một điều kiện OR trong khi **BETWEEN** chỉ ra một điều kiện AND.

Để chọn từ bảng “NhanVien” tên của những người có Tuoi là 22, 28 và 35 có 2 phương pháp như dưới đây. Cả 2 phương pháp đều đưa ra những kết quả giống nhau.

[26] (Chú ý) Nếu các cột có cùng tên, các biến sẽ được sử dụng theo cách chỉ ra dưới đây. Ở đây, biến X được gán đến bảng NhanVien, biến Y gán đến bảng PhongBan.
SELECT Ten, X.Phong, QueQuan, TruongPhong, DiaDiem
FROM NhanVien X, PhongBan Y
WHERE X.Phong = Y.Phong

- SELECT Ten FROM NhanVien WHERE Tuoi IN (22, 28, 35)
- SELECT Ten FROM NhanVien WHERE Tuoi = 22 OR Tuoi = 28 OR Tuoi= 35

Để chọn từ bảng "NhanVien" tên của những người có Tuoi từ 22 đến 28 cũng có 2 phương pháp như dưới đây. Cả 2 phương pháp đều đưa ra những kết quả giống nhau.

- SELECT Ten FROM NhanVien WHERE Tuoi BETWEEN (22, 28)
- SELECT Ten FROM NhanVien WHERE Tuoi >=22 AND Tuoi<=28

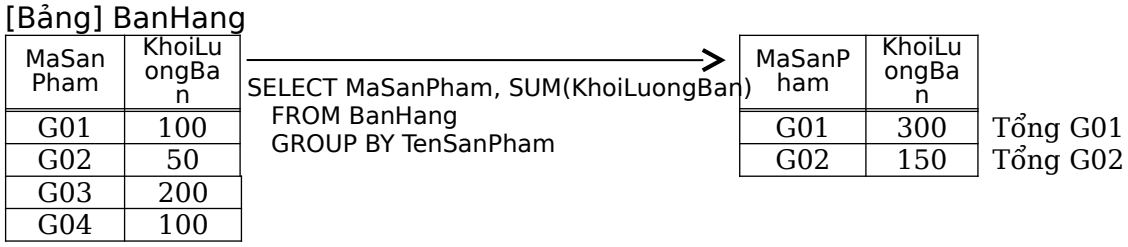
ORDERED BY

ORDERED BY được dùng để chọn ra các dữ liệu và sắp xếp chúng theo thứ tự tăng dần hay giảm dần theo một cột nào đó. Sau đây là một ví dụ trong đó tên được sắp xếp từ bảng "NhanVien" theo thứ tự tăng dần và giảm dần. **ASC** dùng cho thứ tự tăng dần và có thể bỏ qua không cần ghi. Đối với thứ tự giảm dần dùng **DEST**.

- SELECT Ten FROM NhanVien ORDERED BY Ten ASC
- SELECT Ten FROM NhanVien ORDERED BY Ten DESC[27]

Các hàm tập hợp và GROUP BY

Trong bảng "BanHang" dưới đây, chúng ta xem xét một ví dụ của câu lệnh SQL để tìm ra tổng số lượng bán hàng trong các hàng mà có cùng một "MaSanPham". Để tính tổng khối lượng sử dụng hàm tập hợp SUM. Hàm **GROUP BY** hợp nhất tất cả các hàng có cùng giá trị trong 1 cột nào đó. Ở đây , chúng ta hợp nhất dữ liệu bởi MaSanPham. [28]



[27] (Gợi ý) Các tên cột trong "ORDERED BY" hoặc "GROUP BY" phải được chứa trong cột có tên chỉ ra bởi SELECT. Đây là một yêu cầu cú pháp của SQL.

[28] **Hàm tập hợp**: Đây là hàm của phần mềm cơ sở dữ liệu. Nó có thể tìm rất nhiều dữ liệu như tổng cộng và giá trị lớn nhất cho một cột nào đó. Các hàm tập hợp như: SUM (tổng cộng), MAX (giá trị lớn nhất), MIN (giá trị nhỏ nhất), AVG (giá trị trung bình), and COUNT (số lượng).

Các truy vấn con

Chúng ta có thể tạo một câu truy vấn trên một bảng và sau đó sử dụng kết quả của câu truy vấn để tạo ra các câu truy vấn khác. Câu truy vấn đầu tiên gọi là truy vấn con, được thực thi bằng cách dùng IN.

[Bang]DatHang

MaDatHang	MaKhachHang
100	A100
101	B200
102	C300

[Table]ChiTietDatHang

MaDatHang	MaSanPham
100	301
100	302
100	301
100	401
100	402
101	301

[Table]SanPham

MaSanPham	LoaiSanPham
301	television
302	television
401	VCR
402	VCR

Trước tiên chúng ta mô tả phương pháp không dùng truy vấn con. Câu lệnh SQL đưa ra "LoaiSanPham" của sản phẩm được đặt hàng bởi khách hàng có mã A100 từ bảng "DatHang", "ChiTietDatHang" và "SanPham" như sau: [29]

```
SELECT  LoaiSanPham
FROM    DatHang, ChiTietDatHang, SanPham
WHERE   MaKhachHang = 'A100' AND
        DatHang.MaDatHang = ChiTietDatHang.MaDatHang AND
        ChiTietDatHang.MaSanPham = SanPham. MaSanPham
```

Ở đây "MaDatHang" 100 đặt bởi "MaKhachHang" A100 trong bảng "DatHang" được kết nối với "MaDatHang" của bảng "ChiTietDatHang". Kết quả là 5 dòng đầu tiên trong bảng "ChiTietDatHang" với "MaSanPham" 301, 302, 301, 401, và 402 được trả về. Hơn nữa, với việc kết nối với trường "MaSanPham" của bảng "SanPham" thì "LoaiSanPham" cũng được trả về.

Tiếp theo chúng ta sử dụng dạng câu truy vấn con với **IN**.

```
SELECT  LoaiSanPham  FROM  SanPham
WHERE   SanPham. MaSanPham
IN(SELECT ChiTietDatHang.MaSanPham
    FROM  DatHang, ChiTietDatHang
   WHERE  MaKhachHang = 'A100' AND
          DatHang.MaDatHang = ChiTietDatHang.MaDatHang )
```

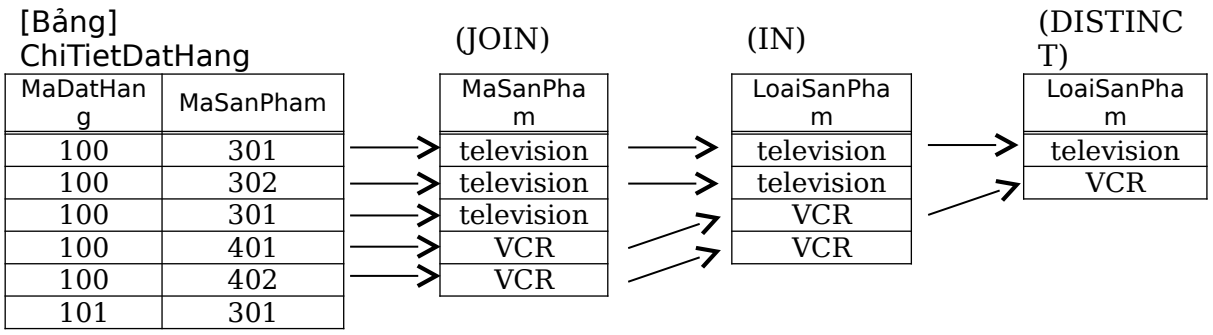
Như đã thấy ở trên, chúng ta có thể tạo ra 1 câu truy vấn và sử dụng kết quả của câu truy vấn đó tạo ra một câu truy vấn khác. [30] Câu truy vấn được chứa trong IN được gọi là câu truy vấn con. Câu lệnh SELECT trong câu truy vấn con được thực thi đầu tiên và kết quả nhận được "ChiTietDatHang.MaSanPham" sẽ kết nối với "SanPham.MaSanPham ". Ở

[29] (Chú ý) Đôi khi EXITST được sử dụng trong các câu truy vấn con. IN và EXIST là các hàm khác nhau nhưng các kết quả thực thi hầu hết giống nhau.
[30] (Chú ý) Chúng ta có thể viết NOT trước IN. Trong trường hợp này NOT IN (câu truy vấn con) đưa ra phủ định kết quả của truy vấn con.

đây, các mã sản phẩm thu được bởi IN là “301, 302, 301, 401, và 402,” nhưng “301” lặp lại nên sự trùng lặp được bỏ qua. Bởi vậy thu được 4 dòng “301, 302, 401, và 402”. [31]
Hơn nữa, các trùng lặp có thể được loại bỏ. Như chỉ ra dưới đây, ta có thể loại bỏ các loại sản phẩm trùng lặp bằng cách dùng DISTINCT trước loại sản phẩm. Television và VCR bị lặp lại nên mỗi loại này được loại bỏ các phần trùng lặp và chỉ xuất hiện 1 lần trong kết quả trả về. [32]

```
SELECT DISTINCT LoaiSanPham
FROM SanPham
WHERE  SanPham. MaSanPham
IN(SELECT ChiTietDatHang.MaSanPham
   FROM  DatHang, ChiTietDatHang
  WHERE  MaKhachHang = 'A100' AND
         DatHang.MaDatHang = ChiTietDatHang.MaDatHang )
```

Bảng dưới đây sẽ mô tả rõ hơn kết quả:



[31] (FAQ) Mỗi kỳ thi thường có nhiều câu hỏi về trích rút dữ liệu từ câu SELECT trong SQL. Cần thông thạo cách sử dụng của SELECT.
[32] (Gợi ý) Chú ý rằng số lượng kết quả trích chọn dữ liệu thay đổi theo điều kiện của WHERE. IN cũng giống như điều kiện OR vì thế các giá trị giống nhau được loại bỏ. DISTINCT cũng loại bỏ các giá trị giống nhau nhưng nó được đưa ra ngay trước tên cột trong SELECT.

Câu hỏi nhanh

Q1 Giải thích vai trò của DDL và DML

Q2 Giải thích hàm con trỏ.

Q3 Cho biết kết quả trả về từ bảng "BangDanhSachSinhVien" sau khi thực hiện câu truy vấn sau:

```
SELECT Ten FROM BangDanhSachSinhVien
WHERE ChuyenNganh = 'Physics' AND Tuoi < 20
```

[Table] BangDanhSachSinhVien

Ten	ChuyenNganh	Tuoi
Paul Newman	Physics	22
John Wayne	Chemistry	20
Tom Hanks	Biology	18
Robert Redford	Physics	19
Clint Eastwood	Mathematics	19

A1 DDL: Ngôn ngữ định nghĩa dữ liệu: Một ngôn ngữ hệ thống sử dụng để định nghĩa các lược đồ dựa trên mô hình dữ liệu.

DML: Ngôn ngữ thao tác dữ liệu: Một ngôn ngữ hệ thống dùng để thao tác các cơ sở dữ liệu được sử dụng bởi người dùng.

A2 Đây là hàm sử dụng khi xử lý các hàng (bản ghi) của một cơ sở dữ liệu sử dụng một ngôn ngữ lập trình. Khi các kết quả truy vấn của DML bao gồm nhiều hàng, ta có thể đọc từng hàng một giống như xử lý các tệp với ngôn ngữ lập trình.

A3 Xem xét câu lệnh SQL chúng ta thấy ý nghĩa của nó là "Lấy từ danh sách sinh viên tên của các sinh viên có chuyên ngành là physics và có tuổi <20". Do vậy, kết quả trả về là "Robert Redford."

5.3 Kiểm soát cơ sở dữ liệu

Mở đầu

Để chắc chắn tính tin cậy của dữ liệu rất nhiều các kiểm soát được áp dụng với một hệ thống cơ sở dữ liệu. Trong một cơ sở dữ liệu phân tán, điều này là rất quan trọng để duy trì tính nhất quán.

5.3.1 Các chức năng kiểm soát cơ sở dữ liệu

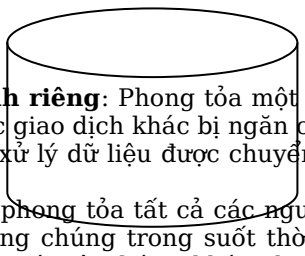
Điểm chính	➢ **Kiểm soát truy cập dành riêng được đưa ra nhằm duy trì tính nhất quán dữ liệu.** ➢ **Các phương pháp phục hồi lỗi bao gồm quay lui (*roll-back*) và quay tiến (*roll-forward*).**

Các chức năng kiểm soát cơ sở dữ liệu bao gồm các chức năng điều khiển truy cập và quản lý chia sẻ tài nguyên. Đồng thời chúng cũng hỗ trợ việc phục hồi dữ liệu khi gặp lỗi.

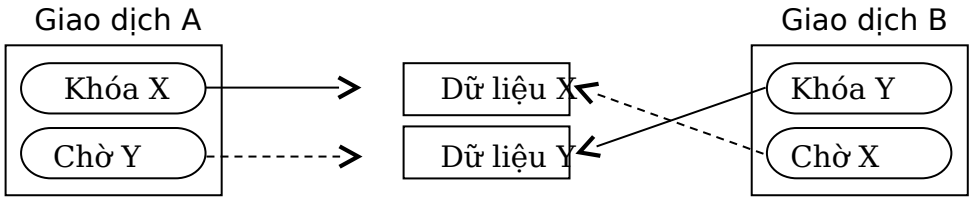
Kiểm soát truy cập

Nói chung, với việc quản lý truy cập nhằm duy trì tính toàn vẹn của một cơ sở dữ liệu ta sử dụng điều khiển truy cập dành riêng (*exclusive access control*) [33] Điều khiển truy cập dành riêng ngăn chặn việc nhiều người dùng truy cập cùng một dữ liệu tại cùng một thời điểm. Qua điều khiển này, nhiều người dùng có thể sử dụng cùng một cơ sở dữ liệu mà không gây ra các xung đột. Tuy nhiên, nếu tất cả các giao dịch (*transactions*) [34] chỉ là các tham chiếu, điều khiển truy cập dành riêng là không cần thiết.

Trong một số trường hợp, kiểm soát truy cập dành riêng có thể gây ra một khóa chết (*deadlock*). Một **khóa chết** là tình trạng trong đó 2 giao dịch đều đang trong tình trạng chờ trạng thái kia giải phóng. Hình vẽ mình họa cho khóa chết như dưới đây. [35]



[33] **Điều khiển truy cập dành riêng**: Phong tỏa một vùng của cơ sở dữ liệu khi nó được cập nhật bởi một giao dịch sao cho các giao dịch khác bị ngăn cấm truy cập trên vùng cơ sở dữ liệu ấy.
[34] **Giao dịch**: Là một đơn vị xử lý dữ liệu được chuyển từ thiết bị đầu cuối đến máy chủ. Nó cũng được gọi là một thông điệp.
[35] (Chú ý)Nếu mỗi giao dịch phong tỏa tất cả các nguồn tài nguyên cần thiết tại thời điêm nó bắt đầu xử lý và không giải phóng chúng trong suốt thời gian xử lý thì có thể tránh được một khóa chết. Hoặc nếu mọi giao dịch có một thứ tự khác nhau của việc phong tỏa, khóa chết cũng có thể tránh.



Quản lý chia sẻ tài nguyên

Có nhiều người dùng truy cập cơ sở dữ liệu thông qua các ứng dụng. Ở đây việc chỉnh sửa, cập nhật dữ liệu cần được hoàn thành bởi chỉ 1 ứng dụng. Vì lý do này chức năng quản lý giao dịch *(transaction management function)* được giới thiệu. Chức năng quản lý giao dịch nhằm duy trì tính nhất quán dữ liệu, tránh các xung đột bằng cách chia dữ liệu thành các đơn vị giao dịch sao cho cơ sở dữ liệu có thể điều khiển được việc cập nhật từ các người dùng khác nhau. Nó bao gồm các đặc điểm **ACID** *(ACID characteristics)*.[36]

Tên		Giải thích
A	Nguyên tố (*Atomicity*)	Không có giai đoạn trung gian khi kết thúc quá trình xử lý. Hoặc tất cả các xử lý đều được hoàn thành hoặc không có gì được thực hiện cả.
C	Nhất quán (*Consistency*)	Dù điều kiện hoàn thành một giao dịch như thế nào, các nội dung của một cơ sở dữ liệu không thể có những mâu thuẫn.
I	Cô lập (*Isolation*)	Các kết quả xử lý không thể khác biệt dù các giao dịch được xử lý đồng thời hay tuần tự
D	Bền vững (*Durability*)	Các kết quả không bị phá hỏng bởi lỗi và các nhân tố khác một khi các giao dịch kết thúc.

Quản lý phục hồi

Có nhiều phương pháp quản lý phục hồi, phụ thuộc vào tình trạng lỗi của cơ sở dữ liệu.

Loại lỗi	Phương pháp phục hồi	Giải thích
Lỗi hệ thống	Khởi động lại hệ thống	Một lỗi hệ thống như các lỗi vật lý (1) Quay lại thời điểm khi mà dữ liệu được sao lưu[37] (2) Ghi lại một cách tuần tự sử dụng các thông tin cập nhật (post-update) trong nhật ký. [38]

[36] (FAQ) Có nhiều câu hỏi thi về ý nghĩa và sự cần thiết của điều khiển truy cập dành riêng. Nhớ rằng điều khiển truy cập dành riêng là một hàm ngăn cấm truy cập vào cùng 1 vị trí (bản ghi) tại cùng một thời điểm. Cũng phải hiểu rằng điều khiển truy cập dành riêng được đưa ra nhằm duy trì tính toàn vẹn dữ liệu.

[37] **Back up**: Sao lưu: Là sự nhân đôi các nội dung của một cơ sở dữ liệu vào trong một đối tượng trung gian như đĩa từ. Việc sao chép được thực hiện tại những khoảng thời gian định kỳ. Thông thường các nội dung sao chép là những dữ liệu ngay trước khởi động hoặc các giá trị ngay trước khi tắt của một hệ thống online. Nếu một hệ thống được thực thi 24h một ngày, việc sao lưu được thực thi khi các giao dịch là ít nhất chẳng hạn như nửa đêm.

[38] **File nhật ký**: Là bản ghi dữ liệu trong đó các điều kiện của cơ sở dữ liệu trước và sau cập nhật được ghi lại mỗi khi các nội dung của cơ sở dữ liệu được cập nhật. Trong một số hệ thống chỉ các

Lỗi giao dịch	Quay lui (*Roll-back*)	Lỗi logic của các thực thi dẫn đến chương trình bị lỗi … (1) Chỉ quay lui lại dữ liệu các lỗi, sử dụng các thông tin trước cập nhật trong nhật ký (2) Tái thực hiện lại giao dịch
Lỗi môi trường	Quay tiến (*Roll-forward)*	Vấn đề với môi trường ghi như đĩa từ (1)Thay thế môi trường ghi (2) Quay lại thời điểm dữ liệu được sao lưu (3) Thực hiện lại tuần tự theo các thông tin trước cập nhật trong file nhật ký.

nội dung trước cập nhật mới được ghi lại.

5.3.2 Các cơ sở dữ liệu phân tán

Điểm chính	➢ **Xác nhận 2 pha là công nghệ duy trì tính toàn vẹn của một cơ sở dữ liệu phân tán** ➢ **Nhân bản là công nghệ nâng cao trình diễn đáp ứng của một cơ sở dữ liệu phân tán**

Cơ sở dữ liệu phân tán là công nghệ lấy nhiều cơ sở dữ liệu khác nhau để trên nhiều máy tính kết nối với một mạng và khiến cho chúng xuất hiện như một cơ sở dữ liệu đơn nhất. Do đó người dùng không nhất thiết phải biết được máy tính nào thực sự có dữ liệu người đó cần.

Xác nhận 2 pha

Xác nhận 2 pha *(Two-phase commitment)* là cơ chế nhằm đảm bảo tính toàn vẹn của một cơ sở dữ liệu phân tán và rất quan trọng khi cập nhật cơ sở dữ liệu.[39]

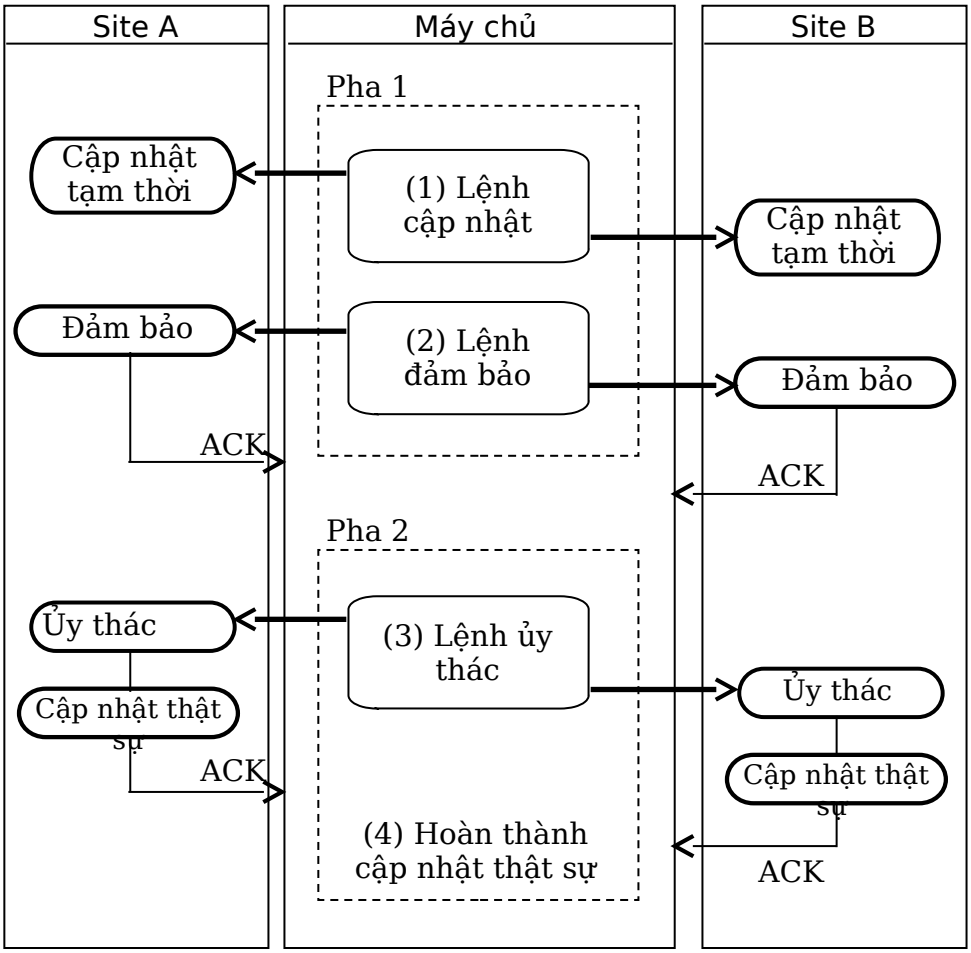
Xác nhận 2 pha có 2 pha khác nhau. Trong pha thứ 1, thành viên yêu cầu sự đồng bộ hóa tạo ra một yêu cầu đến tất cả các thành viên khác về sự đảm bảo cho lệnh cập nhật. Tại thời điểm này, tất cả các thành viên đều bảo lưu.[40] Sau đó mỗi thành viên trả về hoặc xác nhận (*COMMIT*) hoặc quay lui (*ROLLBACK*) [41] cho thành viên yêu cầu đồng bộ hóa. Trong pha thứ 2, thành viên yêu cầu sự đồng bộ hóa xem xét sự phản hồi từ mỗi thành viên và quyết định COMMIT hay ROLLBACK. Đặc biệt, thành viên yêu cầu sẽ lựa chọn ROLLBACK khi thậm chí chỉ có một thành viên trả về ROLLBACK.
Hình vẽ trong trang bên chỉ ra một tiến trình xử lý trong các trường hợp thông thường. Chúng ta sử dụng hình vẽ này để giải thích sự ủy thác 2 pha. “ACK” trong hình vẽ là thông điệp phản hồi chỉ ra sự hoàn thành một cách bình thường.

Site A và Site B là các vị trí đặt cơ sở dữ liệu phân tán. Máy chủ là máy tính điều khiển cơ sở dữ liệu phân tán này. Khi cơ sở dữ liệu được cập nhật, máy chủ chuyển một lệnh cập nhật (1) đến mỗi site. Mỗi site sẽ tiến hành cập nhật tạm thời cơ sở dữ liệu. Đây là một điều kiện cho biết tại đâu cơ sở dữ liệu có thể tiến hành cập nhật tại bất cứ thời điểm nào, nhưng cơ sở dữ liệu chưa tiến hành cập nhật một cách vật lý. Hơn nữa, mỗi site tự nó chuẩn bị cho việc cập nhật và xóa bỏ tại bất cứ thời điểm nào khi mà nó nhận được lệnh đảm bảo (2) từ máy chủ. Đây là pha thứ 1. Nếu có site nào có vấn đề trong pha này việc cập nhật sẽ bị hủy bỏ tại mọi site.

Bước tiếp theo, sau khi chắc chắn rằng các pha đã sẵn sàng cho việc cập nhật (ACK), máy chủ chuyển một lệnh xác nhận (3) cho tất cả các site một cách

[39] **Ủy thác**: Có nghĩa là kết thúc một cập nhật cơ sở dữ liệu. Chỉ sau nó, kết quả sử lý mới được lưu giữ. Khi một ứng dụng thực thi một lệnh COMMIT, việc cập nhật mới kết thúc.
[40] **Tình trạng bảo lưu**: Là tình trạng mà có thể hoàn tất một xử lý hoặc quay lại tình trạng trước.
[41] **Quay lui**: Dừng xử lý và trả về các thông tin liên quan đến thời điểm trước xử lý. Điều này diễn ra khi có một vài sự cố xảy ra trong xử lý giao dịch gây ra việc xử lý không thể hoàn tất một cách bình thường. Quay lui có thể được thực hiện với hệ quản trị cơ sở dữ liệu nhưng cũng có thể được thực thi bởi lệnh ROLLBACK qua một ứng dụng.

tuần tự. Site A với việc nhận được lệnh xác nhận tiến hành cập nhật thật sự và thông báo sự hoàn thành bình thường của xử lý cho máy chủ (ACK). Máy chủ sau đó chuyển lệnh xác nhận sang site B. Tương tự như vậy site B cũng tiến hành việc cập nhật thật sự và thông báo sự hoàn thành bình thường của xử lý cho máy chủ (ACK). Sai đó máy chủ sau đó sẽ xác nhận việc toàn bộ cơ sở dữ liệu được cập nhật thật sự (4). Đây là pha thứ 2. [42]



Nhân bản

Nhân bản là cơ chế tự động phản ánh các nội dung được cập nhật cho các bản sao của cơ sở dữ liệu trên mạng. Mục đích của việc nhân bản là nâng cao sự đáp ứng của quá trình truy cập dữ liệu trong một môi trường cơ sở dữ liệu phân tán.
Các bản sao của dữ liệu chủ được đặt trên các máy chủ khác trên mạng, khi dữ liệu được cập nhật sự thay đổi sẽ tự động được ánh xạ đến các bản sao. Tuy nhiên việc cập nhật của các bản sao không được tiến hành đồng bộ cùng với dữ liệu chủ.
Nói chung, việc cập nhật có thể chỉ được tiến hành trên máy chủ và các bản sao chỉ được ánh xạ nhằm đảm bảo sự toàn vẹn dữ liệu.

[42] (FAQ) Nhiều câu hỏi thi liên quan đến lỗi cơ sở dữ liệu với quay lui (roll-back) và quay tiến (roll-forward). Bạn chắc chắn sẽ giải quyết được nếu bạn hiểu ý nghĩa của quay lui và quay tiến.

Câu hỏi nhanh

Q1 Giải thích vai trò của DDL và DML.

Q2 Giải thích các đặc điểm ACID.

Q3 Liệt kê những thứ cần thiết để khôi phục lại khi một cơ sở dữ liệu gặp lỗi vì lỗi giao dịch. Phương pháp phục hồi gọi là gì?

Q4 Giải thích Ủy thác 2 pha.

A1 DDL: Ngôn ngữ định nghĩa dữ liệu
DML: Ngôn ngữ thao tác dữ liệu

A2 Các đặc tính ACID là khái niệm của việc duy trì tính nhất quán dữ liệu. Thuật ngữ ACID là viết tắt cho Nguyên tố (Atomicity), Nhất quán (Consistency), Cô lập (Isolation), và Bền vững (Durability).

A3 Cần có thông tin trước cập nhật trong tệp nhật ký. Đó gọi là phương pháp quay lui (rollback).

A4 Đây là một cơ chế đảm bảo tính toàn vẹn của một cơ sở dữ liệu phân tán. Pha đầu là pha mà thành viên yêu cầu đồng bộ hóa tạo một yêu cầu đến các thành viên thực thi xin một xử lý cập nhật. Pha thứ 2 là pha mà thành viên có yêu cầu đồng bộ cân nhắc các phản hồi từ mỗi thành viên và quyết định xem ủy thác (commit) hay quay lui (rollback).

Câu hỏi 1

Độ khó: * Tần suất: ***

Q1. Trong số các phép toán sau đây phép toán nào lấy ra vài cột đặc biệt nào đó từ các bảng trong 1 cơ sở dữ liệu quan hệ?

a) Phép kết nối b) Phép chiếu c) Phép chọn d) Phép hợp

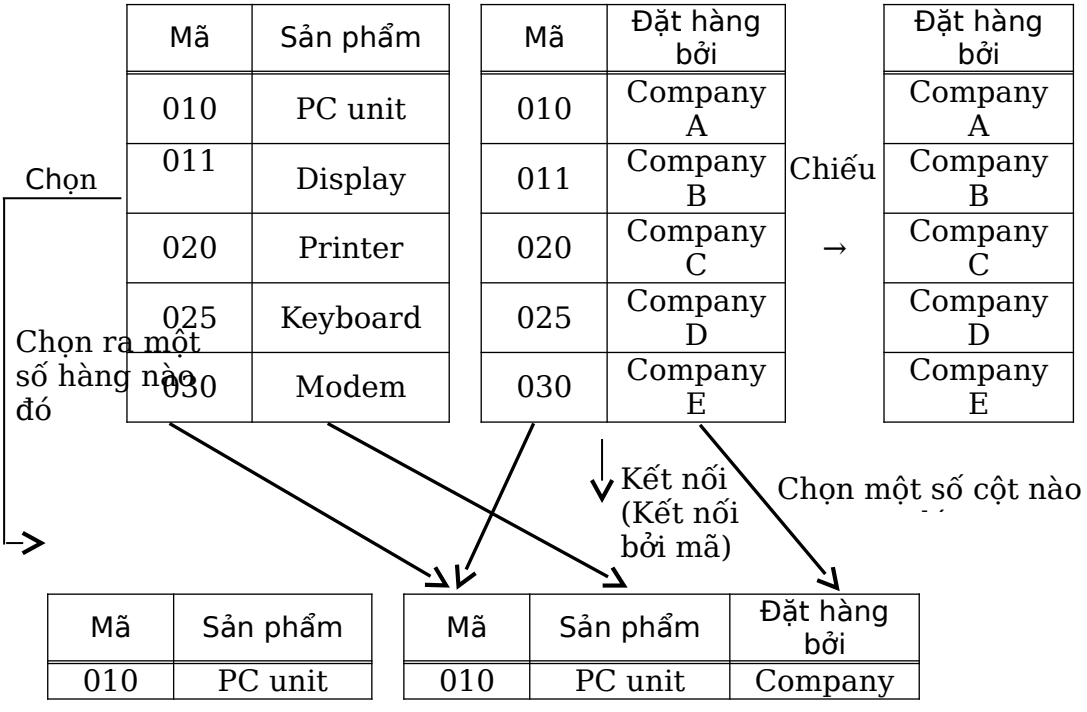
Đáp án câu 1

Đáp án đúng: b

Các thao tác với một cơ sở dữ liệu quan hệ gồm các loại sau:

Các loại phép toán		Contents of operation
Các phép toán quan hệ	Chọn	Chọn ra các hàng thỏa mãn các điều kiện nào đó
	Chiếu	Chọn ra các cột thỏa mãn một điều kiện nào đó
	Kết nối	Kết nối các bảng bằng các cột có cùng giá trị
Các phép toán tập hợp	Hợp	Chọn ra các hàng nằm trong ít nhất một bảng (các hàng trùng lặp chỉ được tính 1 lần)
	Giao	Chọn ra các hàng nằm trong tất cả các bảng
	Trừ	Chọn ra tất cả các hàng nằm trong 1 bảng và không nằm trong bảng khác

Hoạt động của các phép toán quan hệ chỉ ra dưới đây:



020	Printer
030	Modem

		A
011	Display	Company B
020	Printer	Company C
025	Keyboard	Company D
030	Modem	Company E

Câu hỏi 2

Độ khó: ** Tần suất: ***

Q2. Hãy cho biết đâu là dạng chuẩn 3 của bảng "KyNang" chứa các thông tin về kỹ năng nhân viên? Ở đây các trường gạch chân là các khóa chính.

KyNang(NhanVienID, Ten, {MaKyNang, TenKyNang, SoNamKinhNghiem})

Chú ý rằng, cặp ngoặc "{" và "}" cho biết sự lặp lại.

a)

NhanVienID	Ten	MaKyNang	TenKyNang	SoNamKinhNghi

b)

NhanVienID	Ten	MaKyNang	SoNamKinhNghi

MaKyNang	TenKyNang

c)

NhanVienID	MaKyNang	SoNamKinhNghi

NhanVienID	Ten

MaKyNang	TenKyNang

d)

NhanVienID	MaKyNang

NhanVienID	Ten	SoNamKinhNghi

MaKyNang	TenKyNang

Đáp án câu 2

Đáp án đúng: **c**

Bảng thông tin kỹ năng không đi kèm các lời giải thích nhưng chúng ta cần xem xét dự đoán khóa chính. Ở đây có thể cho rằng khóa chính là mã nhân viên (NhanVienID). Từ các phần bị lặp lại của bảng thông tin kỹ năng ta có thể thấy mỗi mã nhân viên có thể ứng với nhiều mã kỹ năng. Điều đó nói lên rằng mỗi nhân viên có thể có nhiều kỹ năng. Do đó số năm kinh nghiệm không thể xác định được trừ khi 2 trường (mã nhân viên và mã kỹ năng) đã biết. Chúng ta có thể thấy rằng một nhân viên không thể có cùng một mã kỹ năng nhiều hơn một lần. Mặt khác như giải thích phía sau thì mã kỹ năng không thể tự mình làm khóa chính.

Đầu tiên, loại bỏ sử lặp lại của "MaKyNang, TenKyNang, SoNamKinhNghiem" trong bảng. Dữ liệu sau đó được chia thành nhiều bản ghi khác nhau, tuy nhiên nếu nhân viên ID tự mình làm khóa chính thì sự trùng lặp có thể xảy ra. Do đó chúng ta cần phải kết hợp nhân viên ID và mã kỹ năng làm thành khóa

chính. Đây là dạng chuẩn 1.

KyNang(NhanVienID, Ten, {MaKyNang, TenKyNang, SoNamKinhNghiem})
→ KyNang (NhanVienID, Ten, MaKyNang , TenKyNang, SoNamKinhNghiem)

Tiếp theo, chú ý rằng tên (Ten) chỉ phụ thuộc hàm vào riêng trường nhân viên ID (phụ thuộc hàm bộ phận vào khóa chính), tên kỹ năng chỉ phụ thuộc hàm vào riêng trường mã kỹ năng (phụ thuộc hàm bộ phận vào khóa chính), và Số năm kinh nghiệm phụ thuộc hàm vào "NhanVienID, MaKyNang" (phụ thuộc hàm đầy đủ vào khóa chính) khi mà một nhân viên có thể có nhiều kỹ năng.

Bây giờ chúng ta cần phải loại bỏ các phụ thuộc hàm bộ phận. Đây là dạng chuẩn 2. Chúng ta gán tên cho từng bảng mới tách ra như giải thích phía dưới.

KyNang (NhanVienID, Ten, MaKyNang , TenKyNang, SoNamKinhNghiem)
→ NhanVien(NhanVienID, Ten)
→ ThongTinKyNang (MaKyNang , TenKyNang)
→ KyNang (NhanVienID, MaKyNang, SoNamKinhNghiem)

Sau sự phân tách trên mỗi bảng chỉ có duy nhất 1 trường không là khóa, do đó không thể có các phụ thuộc hàm bắc cầu. Bởi thế đây cũng là dạng chuẩn 3.

a) Dạng chuẩn 1
b) "Ten" chỉ có thể xác định đơn nhất dựa vào NhanVienID nên đó là một phụ thuộc hàm bộ phận. Đây là dạng chuẩn 1.
c) "SoNamKinhNghiem" phụ thuộc hàm vào "NhanVienID,MaKyNang". Do đó "SoNamKinhNghiem" không phụ thuộc hàm đối với khóa chính "NhanVienID".

Câu hỏi 3

Độ khó: * Tần suất: ***

Q3. Đâu là mô tả thích hợp nhất của khóa chính trong cơ sở dữ liệu quan hệ:

a) Các hàng không thể được tìm ra trừ khi điều kiện tìm kiếm là cho cột chứa khóa chính.
b) Nếu một cột nào đó chứa các giá trị số được gán khóa chính thì sau đó cột đó có không thể được sử dụng trong các phép toán số học.
c) Các hàng với các giá trị khóa chính giống nhau không thể xuất hiện trong cùng 1 bảng.
d) Không thể tạo thành khóa chính gồm có nhiều cột khác nhau.

Đáp án câu 3

Đáp án đúng: c

Trong một cơ sở dữ liệu quan hệ, khóa chính được tạo ra bằng cách kết hợp 1 hoặc nhiều trường nhằm mục đích xác định một hàng (bản ghi). Khóa chính không thể mang các giá trị trùng lặp trong cùng 1 bảng.

a) Các hàng có thể được đọc từ đầu theo thứ tự.

b) Một khóa chính là một trường hoặc một tập các trường (cột) sao cho nó không có sự trùng lặp giá trị của bảng và không xác định rõ các thuộc tính của dữ liệu. Do vậy, dữ liệu có thể được sử dụng cho các phép toán.

c) Một khóa chính có thể tạo thành bằng cách kết hợp nhiều cột khác nhau.

Câu hỏi 4

Độ khó: ** Tần suất: **

Q4. Đâu là mô tả đúng nhất cho các khung nhìn cơ sở dữ liệu quan hệ?

a) Không thể định nghĩa một khung nhìn từ nhiều bảng.
b) Khi một cột được thêm vào một bảng khung nhìn phải được định nghĩa lại.
c) Những người dùng không chỉ biết cấu trúc của khung nhìn mà còn biết cấu trúc của bảng gốc.
d) Các khung nhìn hữu ích trong việc bảo vệ dữ liệu và toàn vẹn dữ liệu khi mà các dữ liệu được sử dụng có thể được hạn chế.

Đáp án câu 4

Đáp án đúng: d

Một khung nhìn (bảng ảo) là sự mô tả cái nhìn cơ sở dữ liệu từ quan điểm của một ứng dụng. Nó là một bảng tách biệt định nghĩa bằng cách kết hợp các khoản mục cần thiết từ các bảng khác nhau hay từ một bảng duy nhất. Với việc chỉ duy nhất các khoản mục cần thiết với người sử dụng được định nghĩa các dữ liệu có thể bị sử dụng sẽ được hạn chế, cho phép bảo vệ và toàn vẹn dữ liệu. Hơn nữa, một khung nhìn xuất phát từ một bảng có thể được cập nhật, nhưng một khung nhìn xuất phát từ nhiều bảng thì không thể cập nhật. Thậm chí nếu một khung nhìn xuất phát từ một bảng đơn lẻ nó không thể được cập nhật nếu sử dụng các điều kiện sau:

- DISTINCT: Trong DISTINCT, các hàng với các giá trị trùng lặp được kết hợp lại thành 1 hàng, vì vậy những hàng ở bảng gốc cần được cập nhật sẽ không thể xác định được.
- Các hàm tập hợp, tính toán: Giá trị thu được bởi các tính toán (ví dụ giá trị thu được bằng phép toán SUM như là tổng cộng) không thể được cập nhật. Trong trường hợp này giá trị gốc nên được cập nhật.
- Các truy vấn con
- GROUP BY, HAVING

Một khung nhìn gọi là một lược đồ ngoài trong một cơ sở dữ liệu quan hệ và một lược đồ con trong cơ sở dữ liệu mạng.

a) Một khung nhìn có thể định nghĩa bằng cách kết hợp nhiều bảng.

b) Một khung nhìn không bị ảnh hưởng khi một cột được thêm vào bảng gốc.

c) Với việc chỉ các khoản mục cần thiết mới được chọn và định nghĩa từ bảng gốc khung nhìn không liên quan đến cấu trúc của bảng gốc.

Câu hỏi 5

Độ khó: * Tần suất: ***

Q5. Câu lệnh SQL nào dưới đây có thể thu được bảng B từ bảng A?

[Bảng] A

MaNhanVien	Ten	MaPhong	Luong
10010	Lucy Brown	101	2,000
10020	Mike Gordon	201	3,000
10030	William Smith	101	2,500
10040	John Benton	102	3,500
10050	Tom Cage	102	3,000
10060	Mary Carpenter	201	2,500

[Bảng] B

MaPhong	MaNhanVien	Ten
101	10010	Lucy Brown
101	10030	William Smith
102	10040	John Benton
102	10050	Tom Cage
201	10020	Mike Gordon
201	10060	Mary Carpenter

a) SELECT MaPhong, MaNhanVien, Ten FROM A GROUP BY MaNhanVien
b) SELECT MaPhong, MaNhanVien, Ten FROM A GROUP BY MaPhong
c) SELECT MaPhong, MaNhanVien, Ten FROM A ORDER BY MaNhanVien, MaPhong
d) SELECT MaPhong, MaNhanVien, Ten FROM A ORDER BY MaPhong, MaNhanVien

Đáp án câu 5

Đáp án đúng: d

Bảng B có thể chọn MaPhong, MaNhanVien, và Ten từ bảng A. Điều này có thể viết như sau:

SELECT MaPhong, MaNhanVien, ten FROM A

Chú ý rằng các bản ghi được sắp xếp theo MaPhong và trong cùng một MaPhong thì sắp xếp theo MaNhanVien. Điều này được thực hiện với câu lệnh sau đây. Nhớ rằng ASC nghĩa là "Theo thứ tự tăng dần"; nếu thứ tự bị bỏ qua thì mặc định sẽ là ASC.

ORDER BY MaPhong, MaNhanVien

Từ đó để thu được bảng B từ bảng A chúng ta sử dụng câu lệnh SQL sau:

SELECT MaPhong, MaNhanVien, Ten *; lấy MaPhong, MaNhanVien, Ten*
FROM *; từ bảng A*
ORDER BY MaPhong, MaNhanVien *; Sắp xếp theo thứ tự tăng dần của mã phòng và mã nhân viên.*

Câu hỏi 6

Độ khó: ** Tần suất: *

Q6. Trong một hệ thống cơ sở dữ liệu phân tán, phương pháp nào sau đây được sử dụng để thăm dò xem nhiều site thi hành một loạt các xử lý giao dịch có thể được cập nhật hay không và có thể thực hiện một xử lý cập nhật cơ sở dữ liệu sau khi xác nhận rằng tất cả các site có thể được cập nhật?

a) Ủy thác 2 pha (*Two-phase commit*)
b) Điều khiển từ chối truy cập (*Exclusive access control*)
c) Quay lui (*Roll-back*)
d) Quay tiến (*Roll-forward*)

Đáp án câu 6

Đáp án đúng: a

Một cơ sở dữ liệu phân tán có khả năng truy cập về lý thuyết như một cơ sở dữ liệu duy nhất dù nó được kết nối với nhiều máy tính trong các vị trí địa lý riêng biệt như các xí nghiệp, công sở và các trung tâm nghiên cứu. Trong một cơ sở dữ liệu phân tán, nếu mọt site kết thúc quá trình cập nhật trong khi các site khác hủy bỏ việc cập nhật dữ liệu sẽ mất tính toàn vẹn. Do vậy, cần có một cơ chế để chắc chắn tính toàn vẹn của các cơ sở dữ liệu tại các nhiều địa điểm.

Ủy thác 2 pha là một cơ chế mà nhờ nó sự toàn vẹn của dữ liệu được đảm bảo khi một cơ sở dữ liệu bị phân tán (ví dụ cơ sở dữ liệu phân tán). Nó có 2 pha. Trong pha thứ nhất một thành viên yêu cầu cập nhật cơ sở dữ liệu sẽ thăm dò các thành viên khác của cơ sở dữ liệu phân tán nếu sự ủy thác được cho phép. Ở đây, mỗi cơ sở dữ liệu phản hồi bằng ủy thác (*COMMIT*) hoặc quay lui (*ROLLBACK*), nhưng tại thời điểm này các cơ sở dữ liệu phân tán kìm giữ việc cập nhật. Tình trạng này (COMMIT có thể được thi hành nhưng việc cập nhật thật sự lại bị kìm giữ) gọi là tình trạng đảm bảo.

Trong pha thứ 2, thành viên yêu cầu kiểm tra nội dung phản hồi của các thành viên khác và chỉ dẫn COMMIT hay ROLLBACK đến mỗi cơ sở dữ liệu. Nếu một trong các cơ sở dữ liệu phản hồi ROLLBACK thì thành viên yêu cầu chuyển ROLLBACK đến tất cả các thành viên. Nói cách khác, nếu tất cả đều phản hồi COMMIT thì sau đó thành viên yêu cầu gửi một lệnh COMMIT đến tất cả các thành viên. Lúc này các cơ sở dữ liệu phân tán được cập nhật thật sự.